Feature-based Modeling
User Guide and Menu Reference

CADDS® 5 15.0

DOC38305-009

Parametric Technology Corporation

Copyright © 2007 Parametric Technology Corporation. All Rights Reserved.
User and training guides and related documentation from Parametric Technology Corporation and its subsidiary companies (collectively "PTC") is subject to the copyright laws of the United States and other countries and is provided under a license agreement that restricts copying, disclosure, and use of such documentation. PTC hereby grants to the licensed software user the right to make copies in printed form of this documentation if provided on software media, but only for internal/personal use and in accordance with the license agreement under which the applicable software is licensed. Any copy made shall include the PTC copyright notice and any other proprietary notice provided by PTC. Training materials may not be copied without the express written consent of PTC. This documentation may not be disclosed, transferred, modified, or reduced to any form, including electronic media, or transmitted or made publicly available by any means without the prior written consent of PTC and no authorization is granted to make copies for such purposes.

Information described herein is furnished for general information only, is subject to change without notice, and should not be construed as a warranty or commitment by PTC. PTC assumes no responsibility or liability for any errors or inaccuracies that may appear in this document.

The software described in this document is provided under written license agreement, contains valuable trade secrets and proprietary information, and is protected by the copyright laws of the United States and other countries. It may not be copied or distributed in any form or medium, disclosed to third parties, or used in any manner not provided for in the software licenses agreement except with written prior approval from PTC.

UNAUTHORIZED USE OF SOFTWARE OR ITS DOCUMENTATION CAN RESULT IN CIVIL DAMAGES AND CRIMINAL PROSECUTION.

For Important Copyright, Trademark, Patent, and Licensing Information: For Windchill products, select About Windchill at the bottom of the product page. For InterComm products, on the Help main page, click the link for Copyright 2007. For other products, select Help > About on the main menu for the product.

UNITED STATES GOVERNMENT RESTRICTED RIGHTS LEGEND
This document and the software described herein are Commercial Computer Documentation and Software, pursuant to FAR 12.212(a)-(b) (OCT'95) or DFARS 227.7202-1(a) and 227.7202-3(a) (JUN'95), and are provided to the US Government under a limited commercial license only. For procurements predating the above clauses, use, duplication, or disclosure by the Government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software Clause at DFARS 252.227-7013 (OCT'88) or Commercial Computer Software-Restricted Rights at FAR 52.227-19(c)(1)-(2) (JUN'87), as applicable. 02202007

Parametric Technology Corporation, 140 Kendrick Street, Needham, MA 02494 USA

Table of Contents

Preface

Related Documents ____ xi
Book Conventions ____ xii
Window Managers and the User Interface ____ xii
Online User Documentation ____ xiii
Online Command Help ____ xiv
Printing Documentation ____ xiv
Resources and Services ____ xiv
Documentation Comments ____ xv

Introduction to Feature-based Modeling

Overview of Feature-based Modeling ____ 1-2
Why use Feature-based Modeling? ____ 1-2
Object-oriented Methodology ____ 1-2

Concepts and Definitions ____ 1-3
Feature Instances and Workpieces ____ 1-3
Sample Session: Inserting an Instance of a Boss ____ 1-3
Attributes ____ 1-4
Definitions and Inheritance ____ 1-4
Representations and Teach Feature Sessions ____ 1-4
Attribute Association ____ 1-4
Libraries of Features ____ 1-5
System Features and User-defined Features ____ 1-5
Browsing through Lists ____ 1-5

Inserting an Instance

Overview of Inserting Instances _____ 2-2
Inserting Feature _____ 2-4
Viewing a Feature Before Inserting It _____ 2-4
Displaying a Feature While Inserting It _____ 2-6
Inserting a Feature Using the Normal Option _____ 2-7
Rotating an Inserted Feature _____ 2-7
Removing an Instance _____ 2-8
Description of Options _____ 2-8
Representation Command Files _____ 2-9
What You Must Specify _____ 2-9
Attributes of Size _____ 2-9
Application _____ 2-10
Methods of Applying an Instance to a Workpiece _____ 2-12
Exit and Entry Faces _____ 2-12
Example _____ 2-12
Extend _____ 2-13
Example _____ 2-13
Merge _____ 2-13
Boolean _____ 2-14
Extended Boolean _____ 2-14
Example _____ 2-14
Considerations Regarding Booleans and Local Operations _____ 2-15
NoApply _____ 2-15
System Features _____ 2-16
SYSLIB _____ 2-16
The Virtual Library _____ 2-16
Same Feature Name Exists Across Libraries _____ 2-17
Library Options _____ 2-18
Setting the Tolerance Value for System Library Features _____ 2-19
Categories of System Features _____ 2-21
Additive Features _____ 2-21
Example _____ 2-21
Subtractive Features _____ 2-22
Example _____ 2-22

Thru Holes and Cutouts ____ 2-23
Flats and Grooves ____ 2-24
Full and Partial Slots ____ 2-26
System Feature Options from the Features Task Set ____ 2-28
Description of Options ____ 2-29
Sample Session: Inserting an Instance ____ 2-38
Instance Information Retrieval: The Verify Feature Option ____ 2-43
When Instance is Active ____ 2-43
Entering Conditional Expressions ____ 2-44
When Class is Active ____ 2-46
Instance Modification: The Change Feature Option ____ 2-47
Selecting an Instance ____ 2-47
Choosing and Modifying Attributes ____ 2-49

Creating a Feature

Overview of User-defined Features ____ 3-2
Modifying a Definition ____ 3-2
Modifying a Representation ____ 3-3
Creating a Definition ____ 3-4
Overview of the Process ____ 3-4
Entering Attributes ____ 3-4
Options Related to Creating a Definition ____ 3-5
Description of Options ____ 3-6
You Can Create a Definition in the LDM ____ 3-8
Library Options in the LDM ____ 3-9
Multiple Inheritance ____ 3-9
Basic Concept ____ 3-9
Conflicts across Superclasses ____ 3-10
Utilizing Inheritance ____ 3-11
The System Superclass Feature ____ 3-11
What is the Class Feature? ____ 3-11
Changing the Default ____ 3-11
If Feature is not an Appropriate Superclass ____ 3-11
The System Superclass Primitive ____ 3-12

Creating a Representation ____ 3-13
Overview of the Process ____ 3-13
Associating Attributes to a Representation ____ 3-13
User Dimensions and Constraints ____ 3-14
Features Having Parametrically Controlled Groups of Entities 3-14
Exiting a Teach Feature Session and Locking ____ 3-15
Options Related to Creating a Representation ____ 3-16
Teach Feature Options within the LDM ____ 3-16
Teach Feature ____ 3-16
Options within a Teach Feature Session ____ 3-18
Delete Feature ____ 3-18
Sample Session: Creating a Feature ____ 3-19
The Definition ____ 3-19
The Representation ____ 3-21
The Rotational Sweep ____ 3-21
The User Dimensions ____ 3-23
Association of the Attributes ____ 3-24
The Constraints ____ 3-25
Model Regeneration ____ 3-26
Orientation and Origin ____ 3-26

Using Properties

Concepts ____ 4-2
What Are Properties? ____ 4-2
Why Use Properties? ____ 4-2
How Do You Use Properties? ____ 4-2
Creating a Property ____ 4-3
Modifying a Class ____ 4-3
Property Attributes ____ 4-3
Inheritance ____ 4-4
Property Libraries ____ 4-4
The Define Property Property Sheet ____ 4-5
Sample Session ____ 4-6
Deleting a Property ____ 4-12
Example ____ 4-12
What If Instances Exist? ____ 4-12

What About Superclass? 4-13
Sample Session 4-13

Inserting a Property 4-14
Attaching a Property Instance 4-14
You Cannot See a Property Instance 4-15
Assigning Attribute Values 4-15
Tagnames 4-15
The Insert Property Property Sheet 4-16
Sample Session 4-17

Selecting a Property Instance 4-19
Options Requiring Instance Selection 4-19
Property Instance Selection Options 4-20
More on Selecting by Condition 4-20

Changing Property Attribute Values 4-22
Select Instance by Tagname 4-22
The Change Property Property Sheet 4-22
Sample Session 4-23

Removing a Property Instance 4-25
The Remove Property Property Sheet 4-25
Sample Session 4-26

Making Attachments 4-27
Multiple Instances and Attachments 4-27
If You Change an Attached Instance 4-27
The Attach Property Property Sheet 4-28
Sample Session 4-29
If You Delete An Attached Entity 4-33

Detaching a Property Instance 4-34
If You Detach All Attachments 4-35
Multiple Detachments 4-35
The Detach Property Property Sheet 4-35
Sample Session 4-36

Verifying a Property 4-41
The Verify Property Property Sheet 4-41
Sample Session 4-42

Taxonomy of System Features

Overview _____ A-2
Advantages of Classifying Features _____ A-2
System Features _____ A-2
Form Features Information Model _____ A-3
Form Features _____ A-3
Volume Features _____ A-3
Additive Features _____ A-3
Subtractive Features _____ A-4
System Features _____ A-5
Tabs _____ A-5
Bosses _____ A-5
Circular Bosses _____ A-5
Counterbores _____ A-5
Countersinks _____ A-5
Pockets _____ A-5
Grooves _____ A-6
Flats _____ A-6
Slots _____ A-6
Cutouts _____ A-6
Diagrams of Form Features and System Features _____ A-7
Form Features Diagrams _____ A-7
System Features Diagrams _____ A-7

Command File for Hole_ConicalBottom

Inserting the Rotational Sweep _____ B-2
Inserting the User Dimensions _____ B-2
Associating the Attributes _____ B-2
Setting Constraints _____ B-2
Defining an Orientation _____ B-3
Exiting the Part _____ B-3

Troubleshooting

Messages within an Application Session _____ C-2

Messages for Modeling with Features C-3
Messages for Defining Features C-4
Messages for Teach Feature Definitions C-6

Setup Procedures

Installing and Setting Up the Features Database D-2
The Features Database D-2
Features Process D-2
Features Libraries D-2
Creating a Local Features Database Manually D-4
Installing the Local Features Database D-4
Testing the Features Database Setup D-6
Creating a Shared Features Database D-6
Advantages of a Shared Database D-6
Set up the Client Systems D-7
Test the Features Database Setup D-9
Features Libraries on Client Systems D-9
Example of Cross Mounting D-10
Starting the Features Process D-10
If the Process Does Not Start D-11
Releasing the Port D-12
Choosing a New Port D-12
SunOS and HP D-12
Solaris D-12
Accessing a Local Features Database D-13
Switching Features Databases D-14
Copying a Features Library D-15
Copying a Features Library to Another Library D-15
Copying a Features Library from LDM D-16
Copying a Features Library To Tarfile or Tape D-16
Copying a Features Library from Tarfile or Tape D-17
Backing Up Features Libraries D-18
Updating the LDM with a New Revision D-19

Preface

Feature-based Modeling User Guide and Menu Reference assumes that you are familiar with parametric modeling functionality as described in the *Parametric User Guide and Menu Reference*.

This document describes how to perform the following tasks:

- Basic concepts and terminology of feature-based modeling
- How to insert a feature
- How to create a feature
- Feature-related menus
- Feature-based properties

Related Documents

The following documents may be helpful as you use *Feature-based Modeling User Guide and Menu Reference*:

- *Introduction to CADDS 5*
- *Installing CADDS 5*
- *Managing CADDS 5*
- *Parametric Modeling User Guide and Menu Reference*
- *Customizer and Icon Editor User Guide*

Book Conventions

The following table illustrates and explains conventions used in writing about CADDS applications.

Convention	Example	Explanation
Menu selections and options	List Section option, Specify Layer field	Indicates a selection you must make from a menu or property sheet or a text field that you must fill in.
User-selected graphic location	X, d_1 or P1	Marks a location or entity selection in graphic examples.

User input in CADDS text fields and on any command line	`cvaec.hd.data.param`
	
	`tar -xvf /dev/rst0`
		Enter the text in a CADDS text field or on any command line.

System output	`Binary transfer complete.`	Indicates system responses in the CADDS text window or on any command line.
Variable in user input	`tar -cvf /dev/rst0` filename	Replace the variable with an appropriate substitute; for example, replace filename with an actual file name.
Variable in text	tagname	Indicates a variable that requires an appropriate substitute when used in a real operation; for example, replace tagname with an actual tag name.
CADDS commands and modifiers	INSERT LINE TANTO	Shows CADDS commands and modifiers as they appear in the command line interface.
Text string	`"SRFGROUPA"` or `'SRFGROUPA'`	Shows text strings. You must enclose text string with single or double quotation marks.
Integer	n	Supply an integer for the *n*.
Real number	x	Supply a real number for the *x*.
`#`	`# mkdir /cdrom`	Indicates the root (superuser) prompt on command lines.
`%`	`% rlogin` remote_system_name `-l root`	Indicates the C shell prompt on command lines.
`$`	`$ rlogin` remote_system_name `-l root`	Indicates the Bourne shell prompt on command lines.

Window Managers and the User Interface

According to the window manager that you use, the look and feel of the user interface in CADDS can change. Refer to the following table:

Look and Feel of User Interface Elements

User Interface Element	Common Desktop Environment (CDE) on Solaris and HP	Window Manager Other Than CDE on Solaris, HP, and Windows
Option button	ON — Round, filled in the center OFF — Round, empty	ON — Diamond, filled OFF — Diamond, empty

Look and Feel of User Interface Elements

User Interface Element	Common Desktop Environment (CDE) on Solaris and HP	Window Manager Other Than CDE on Solaris, HP, and Windows
Toggle key	ON — Square with a check mark OFF — Square, empty	ON — Square, filled OFF — Square, empty

Online User Documentation

Online documentation for each book is provided in HTML if the documentation CD-ROM is installed. You can view the online documentation in the following ways:

- From an HTML browser
- From the Information Access button on the CADDS desktop or the Local Data Manager (LDM)

Please note: The LDM is valid only for standalone CADDS.

You can also view the online documentation directly from the CD-ROM without installing it.

From an HTML Browser:

1. Navigate to the directory where the documents are installed. For example,

   `/usr/apl/cadds/data/html/htmldoc/` (UNIX)

   Drive:`\usr\apl\cadds\data\html\htmldoc\` (Windows)
2. Click `mainmenu.html`. A list of available CADDS documentation appears.
3. Click the book title you want to view.

From the Information Access Button on the CADDS Desktop or LDM:

1. Start CADDS.
2. Choose Information Access, the i button, in the top-left corner of the CADDS desktop or the LDM.
3. Choose DOCUMENTATION. A list of available CADDS documentation appears.
4. Click the book title you want to view.

From the Documentation CD-ROM:

1. Mount the documentation CD-ROM.
2. Point your browser to:

   CDROM_mount_point`/htmldoc/mainmenu.html` (UNIX)

CDROM_Drive:`\htmldoc\mainmenu.html` (Windows)

Online Command Help

You can view the online command help directly from the CADDS desktop in the following ways:

- From the Information Access button on the CADDS desktop or the LDM
- From the command line

From the Information Access Button on the CADDS Desktop or LDM:

1. Start CADDS.
2. Choose Information Access, the i button, in the top-left corner of the CADDS desktop or the LDM.
3. Choose COMMAND HELP. The Command Help property sheet opens displaying a list of verb-noun combinations of commands.

From the Command Line: Type the exclamation mark (!) to display online documentation before typing the verb-noun combination as follows:

```
#01#!INSERT LINE
```

Printing Documentation

A PDF (Portable Document Format) file is included on the CD-ROM for each online book. See the first page of each online book for the document number referenced in the PDF file name. Check with your system administrator if you need more information.

You must have Acrobat Reader installed to view and print PDF files.

The default documentation directories are:

- `/usr/apl/cadds/data/html/pdf/`doc_number`.pdf` (UNIX)
- CDROM_Drive:`\usr\apl\cadds\data\html\pdf\`doc_number`.pdf` (Windows)

Resources and Services

For resources and services to help you with PTC (Parametric Technology Corporation) software products, see the *PTC Customer Service Guide*. It includes instructions for using the World Wide Web or fax transmissions for customer support.

Documentation Comments

PTC welcomes your suggestions and comments. You can send feedback electronically to `doc-webhelp@ptc.com`.

Chapter 1

Introduction to Feature-based Modeling

This chapter introduces you to the basic concepts and terminology of feature-based modeling.

- Overview of Feature-based Modeling
- Concepts and Definitions

Overview of Feature-based Modeling

Essentially, feature-based modeling in the Parametric environment of CADDS involves inserting intelligent geometry into a model.

Why use Feature-based Modeling?

Using feature-based modeling includes a number of advantages. You can:

- Insert intelligent parametric geometry into a model without having to visualize, then create, the intermediate construction geometry.
- Associate a variety of information to portions of a model.
- Create a set of features with attributes unique to your design requirements.

Object-oriented Methodology

Feature-based modeling uses object-oriented methodology. While knowledge of object-oriented design (OOD) concepts is helpful, it is not required to use feature-based modeling.

Concepts and Definitions

The following sections describes the concepts of feature-based modeling.

Feature Instances and Workpieces

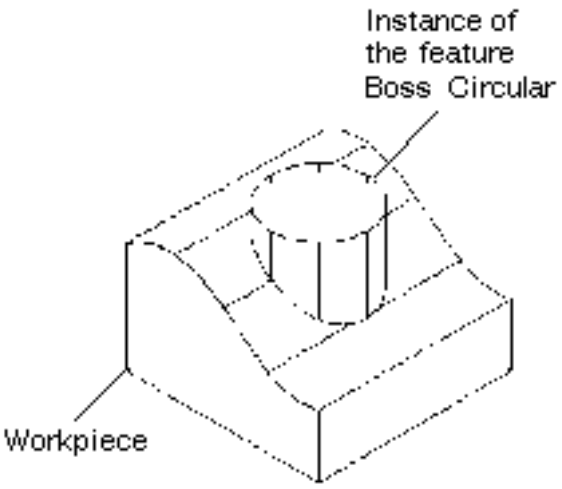
Feature-based modeling allows you to insert features (such as a hole, a boss, or a slot) into a model. Each occurrence of a feature in a model is called an instance. The object within a model, into which, an instance is inserted is called a workpiece**.**

Sample Session: Inserting an Instance of a Boss

Consider the following example. Your current part contains a solid to which you want to add a boss. Using feature-based modeling, you can proceed as follows:

1. Select the feature called Boss_Circular.
2. Assign values to its attributes, that is, topdiameter, bossheight, draft angle.
3. Insert one or more instances.
4. Parametrically change the attribute values of an instance(s).
5. Obtain a report (verify) on an instance.

The following illustration shows an instance of the feature Boss_Circular inserted into a workpiece.



Attributes

An attribute is a characteristic of a feature. Attributes are either static or dynamic. The value of a static attribute is assigned during the definition of the feature. You cannot change that value when you insert an instance. Static attributes are often referred to as class attributes. You can change the default value of a dynamic attribute when inserting an instance of that feature. Dynamic attributes are often referred to as instance attributes.

An attribute can describe either geometric or nongeometric data. For example, one attribute can specify a hole's diameter and another can specify with what material the hole is to be plated. In such a case the diameter attribute describes geometric data and the plating material attribute describes nongeometric data.

Definitions and Inheritance

A definition is a set of attributes that characterizes a particular feature. These attributes are associated to the parameters of a geometric representation. For example, the definition for a hole feature can contain the attributes radius and depth. A definition does not contain any parametric history or geometry.

Subclasses and Superclasses:

A definition can inherit attributes from other definitions. Definitions utilizing inheritance are categorized as either subclasses or superclasses. A superclass is a definition whose attributes are referenced by another definition called a subclass.

Multiple Inheritance:

One superclass may have multiple subclasses and one subclass may have multiple superclasses. The latter concept is called multiple inheritance. For more information on multiple inheritance see "Multiple Inheritance" on page 3-9.

Representations and Teach Feature Sessions

A representation is essentially a model containing the parametric history captured during a special session called a Teach Feature session, or Teach session. In this session you create the geometry for the feature. The resulting parametric history is captured and stored in the representation.

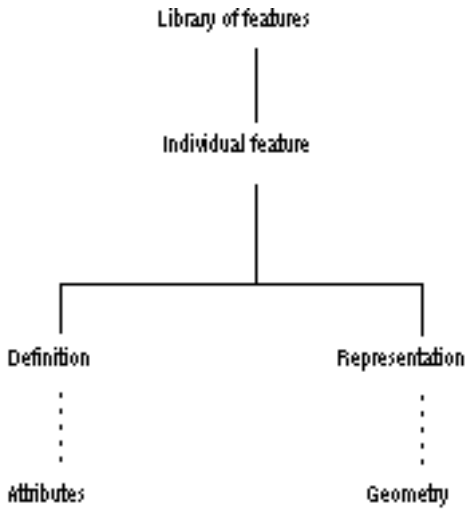
Attribute Association

During a Teach Feature session you associate the attributes of a definition to the representation of that definition. You do so by using the Associate Attribute option which allows you to select a particular option and associate it to a particular parametric dimension, label, user-dimension, or geometric entity by means of a graphical selection.

Libraries of Features

A library contains a collection of features. The following illustration shows:

- A library of features
- A given feature within that library
- The contents of the individual feature
- The contents of each file type within a given individual feature library as indicated by the dotted lines



System Features and User-defined Features

A library called SYSLIB contains a set of system features. The taxonomy of these system features is based on the Product Data Exchange Standard (PDES), which is outlined in Appendix A, “Taxonomy of System Features.” You can select and insert one of these system features. You can also define your own feature. Doing so involves a multistep process that is described in Chapter 3, “Creating a Feature.”

Browsing through Lists

Browsing functionality allows you to browse through lists of libraries, features, and attributes. Menus provide lists for a set of items from which you can make a selection.

Chapter 2

Inserting an Instance

This chapter explains how to insert an instance of a feature.

- Overview of Inserting Instances
- Methods of Applying an Instance to a Workpiece
- System Features
- Categories of System Features
- Sample Session: Inserting an Instance
- Instance Information Retrieval: The Verify Feature Option
- Instance Modification: The Change Feature Option

Overview of Inserting Instances

Select the area shown in the following illustration to display the task set from which you can access the Features menu.

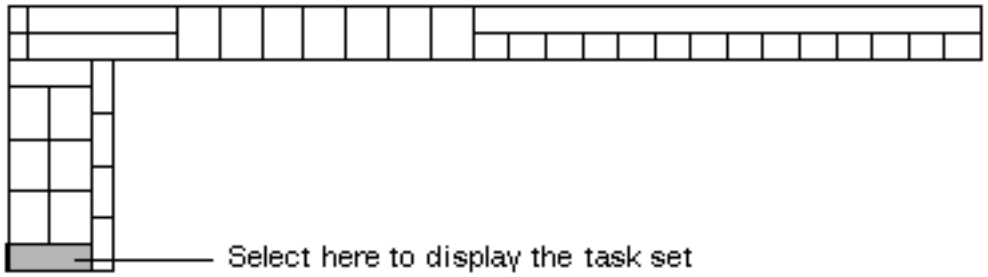
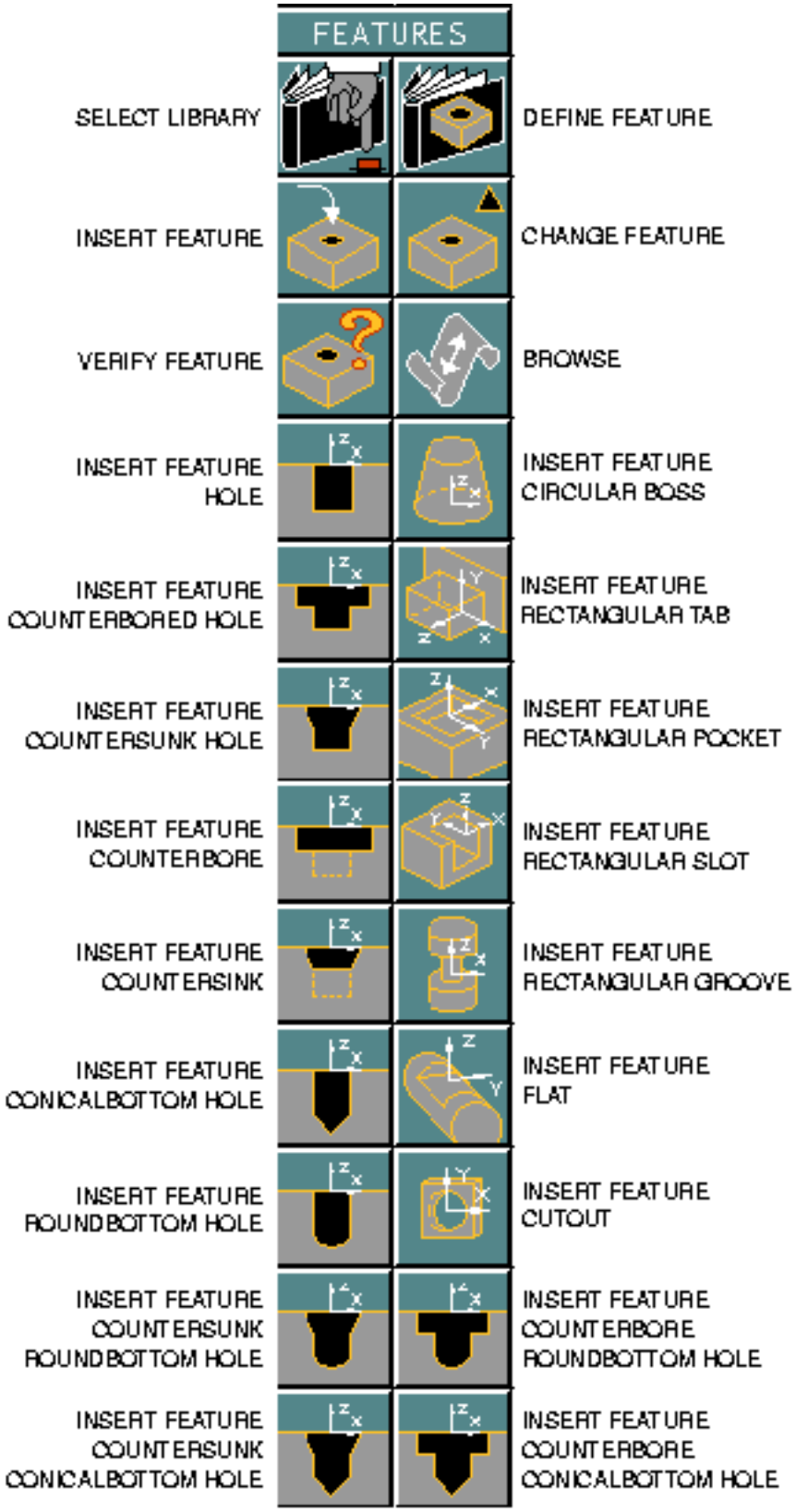
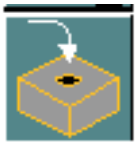


Figure 2-1 The Features Task Set

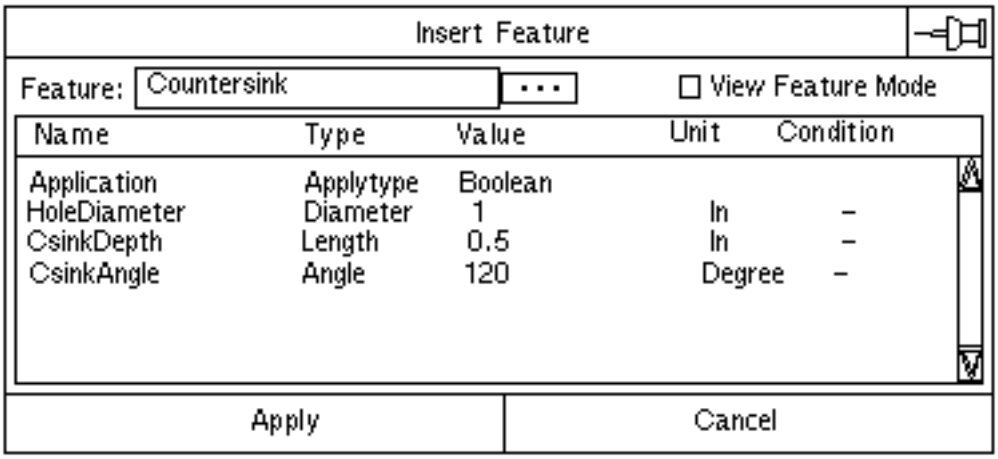


Inserting Feature



Choose the Insert Feature icon on the task set, to get the Insert Feature property sheet, which appears as shown in the following illustration. The Feature field accesses a list of user-defined feature definitions. The set of fields (Name, Type, Value) for each attribute allows you to enter a new value for each attribute. View Feature Mode allows you to view a feature before actually inserting it. It uses the default attribute values of the feature when creating a feature view. Note that the Insert Feature property sheet allows you to select within only SYSLIB and the current user library. Use the Select Library option from the Features Task Set to change the current user library.

Figure 2-2 Insert Feature Property Sheet

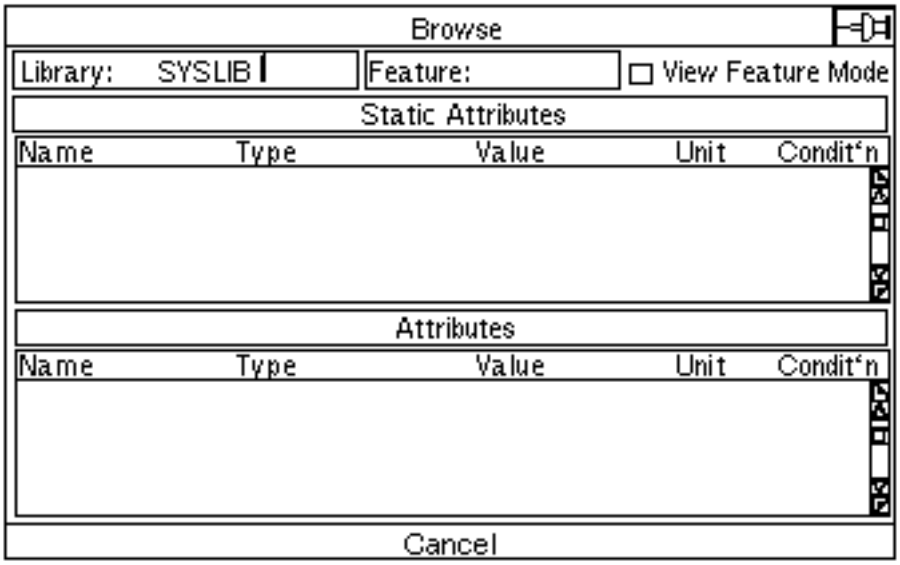


Please note: To insert a user-defined feature, choose the Insert Feature option and specify a name.

Viewing a Feature Before Inserting It

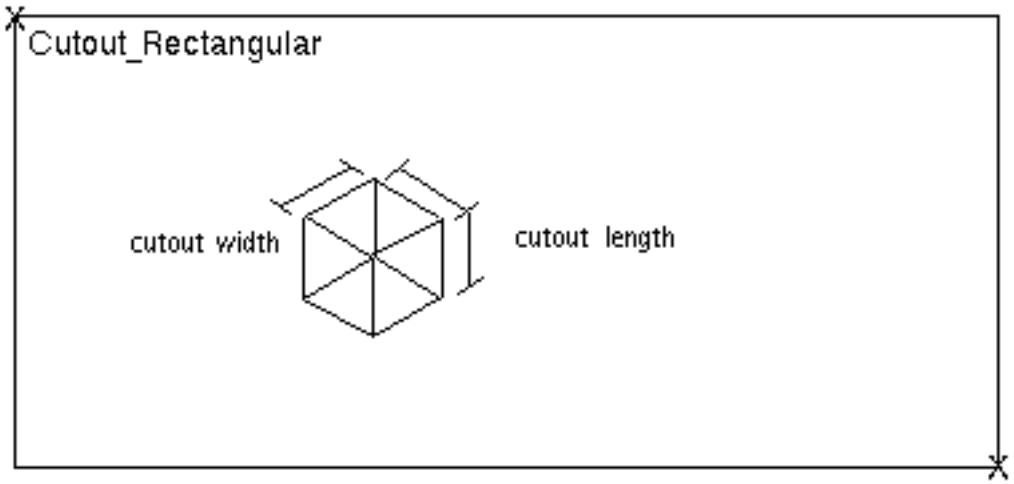
If you want to view a feature before inserting it, you can do so by selecting View Feature Mode from either the Insert Feature property sheet or the Browse property sheet. The Browse property sheet allows you to browse SYSLIB and any user library.

Figure 2-3 Browse Property Sheet



View Feature Mode displays a graphical representation of the feature in the display area that you define by locating two points. The following illustration shows a View Feature Mode display of the system feature Cutout_Rectangular. The two location(X) marks represent the two points that define the size and shape of the display area.

Figure 2-4 View Feature Mode of Cutout_Rectangular Feature



You can:

- Use the Interactive View Controls toolbar to manipulate the graphical representation of the model.
- Locate the view of a feature across multiple views.
- Select multiple features for viewing. The current feature display is replaced with each successive selection.
- Change the size or shape of the display area by selecting two corner points.

Choose Apply on the Insert Feature property sheet. The Insert Feature pulldown menu appears as shown below.

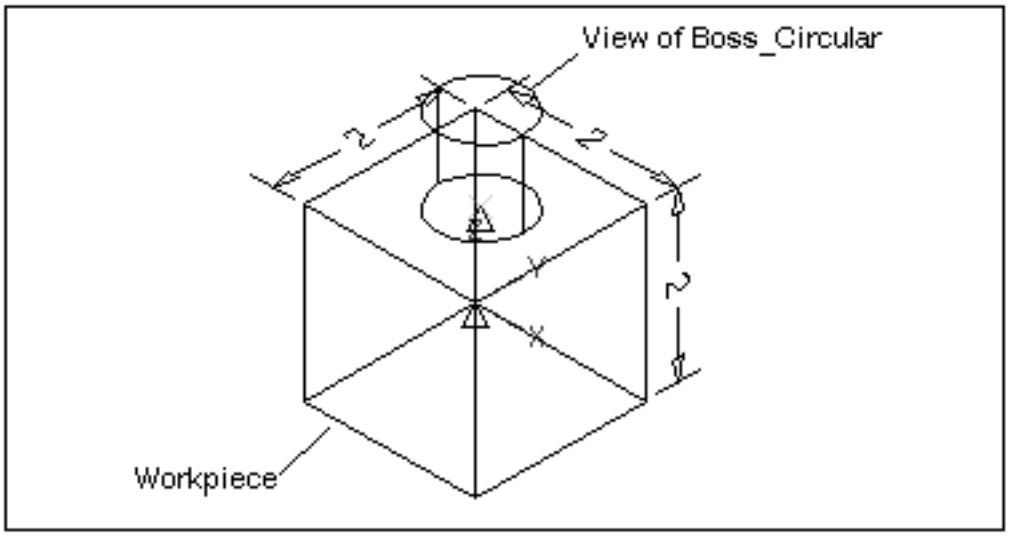
Figure 2-5 Insert Feature pulldown menu

Displaying a Feature While Inserting It

While you insert a feature, you can use the Display option on the Insert Feature pulldown menu shown previously to view that feature. The Display option differs from View Feature Mode as Display allows you to view a feature using the attribute values you have specified. Note that the Display option is also included on each system feature property sheet.

The following illustration demonstrates how the Display option can be used to provide a display of a feature relative to the workpiece. The view shows the system feature Boss_Circular given a specified orientation and origin.

Figure 2-6 Using Display option to view the feature relative to workpiece



By specifying new attribute values and choosing the Display option again, you can get an updated view of the feature.

Inserting a Feature Using the Normal Option

The Normal option on the Insert Feature pulldown menu allows you to insert a feature or multiple copies of the feature using a built-in CPL (construction plane), which has:

- ORIGIN of the given location projected normally to the entry face
- Z-axis oriented normally to the outside of the entry face
- X-axis oriented along the u derivative of the entry face
- Y-axis oriented with respect to the X and Z axes

At the end of this process, the built-in CPLs are deleted.

You first need to select the entry face and the location(s) for this option.

Rotating an Inserted Feature

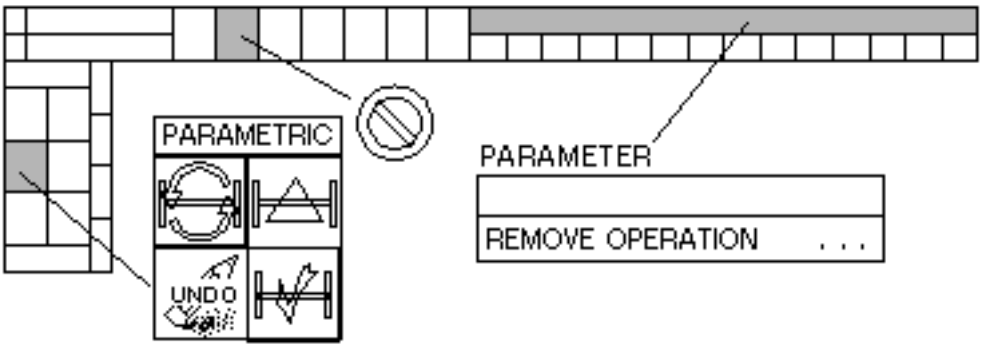
The Rotate option on the Insert Feature pulldown menu allows you to rotate the feature by 90 degrees about the Z axis while inserting it. Choose this option as many times as you want the feature to rotate by 90 degrees.

You need to choose the Normal option before this option.

Removing an Instance

Inserting an instance is an event captured by the parametric history of the current part. Thus you can undo (remove from the parametric history) an instance insertion using the Undo operation. You can delete an unapplied instance (a feature that has been inserted using the NoApply option) or the entire workpiece using the Delete Entity option. You can also use the Remove Operation option from the Parametric menu from the top bar to remove a specified number of events.

Figure 2-7 Options used to remove an Feature Instance

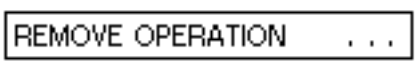


Description of Options

- Undo Operation

Choose the Undo icon from the Environmental Globals area. The Undo Operation property sheet appears. Specify the index number of parametric operations you want undone. (Inserting an instance is a parametric operation.) For Example, if you enter 3, the system removes the last three operations from the parametric history.

- Remove Operation



Choose the Remove Operation option from the Parameter menu of the top bar. Specify the event from the parametric history you want to remove. You can:

- Remove the last event
- Remove the last specified number of events
- Choose a specific event from the displayed list of parametric events

- Delete Entity

Deletes the geometry modified by an instance. That is, it deletes the workpiece. The Delete Entity option also deletes an instance that was not applied (the NoApply option).

Representation Command Files

If, before inserting an instance, you want to know how that feature was taught, review its representation command file. That command file is in the directory:

```
cadds/data/features/cmd/
```

All command files have a `.cmd` suffix and contain all the commands used to create the representation.

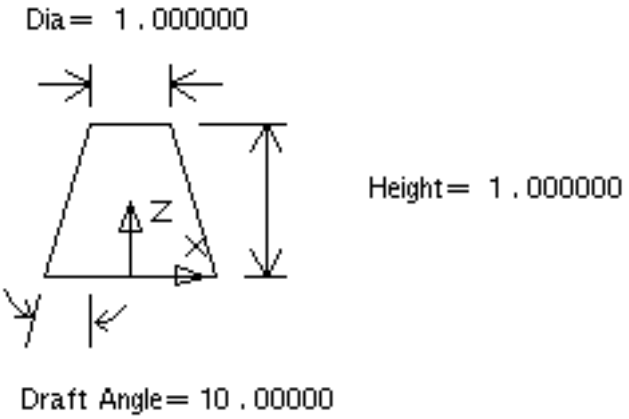
What You Must Specify

An instance is an occurrence of a feature in a model. When you insert an instance, you must assign values to that feature's attributes. When inserting a feature, you must specify a value for some or all of the following attributes:

Attributes of Size

Specify a real number value. Examples of such attributes are the radius of a hole, the height of a boss, and the tapered angle of a cutout. You can accept the default values or enter new values using the calculator that appears when you choose a particular attribute of size.

Example: The following illustration shows the attributes of size of the system feature Boss_Circular. When you insert Boss_Circular you can change the default values of the attributes TopDiameter, BossHeight, and DraftAngle. You can do so by choosing an attribute and using the calculator that appears to enter a new value.



Application

Specify an operation type or application, with which the system applies an instance to the workpiece. You can choose among four applications: Extend, Extended Boolean, Boolean, and Merge. Each feature has a default application. You can alternately choose NoApply. In the case of NoApply the feature is inserted without conducting an operation. First choose an application, then specify these items:

- Workpiece

  The object within a model into which the feature is to be inserted. For example, if you want to insert a hole feature into a block, that block is the workpiece. Select the workpiece graphically. You must select a workpiece for the Boolean and Extended Boolean applications.

- Face attributes

  The face entities within the workpiece into which the instance is to be inserted. You must choose the face attributes for Extend, Merge, and Extended Boolean applications.

- Orientation

  Orients the instance relative to the Cplane you select. The default orientation is relative to the active Cplane.

- Origin

  A point in a model where an instance is inserted by its origin. For a system feature, the instance origin is indicated graphically in its respective property sheet. The default origin is the [0,0,0] of the selected orientation. You can optionally specify multiple locations and hence multiple instances.

Methods of Applying an Instance to a Workpiece

You can choose from the following options when specifying an application:

- Merge
- Extend
- Boolean
- Extended Boolean
- NoApply

Please note: Merge and Extend are local operations. For more information on local operations and Booleans see the *Parametric Modeling User Guide and Menu Reference.*

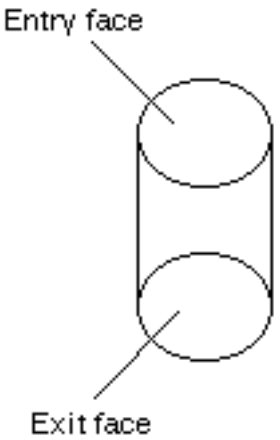
Exit and Entry Faces

When specifying an application you should understand the terms entry face and exit face.

Example

Consider the system feature Hole_Through whose definition includes the attributes EntryFace and ExitFace. The representation of Hole_Through is a cylinder.

As is shown by the following illustration, the attribute EntryFace is associated to a particular face of the Hole_Through's representation as is its ExitFace.



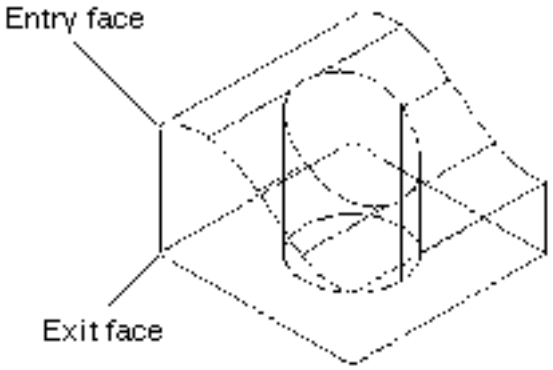
Extend

An Extend application is a local operation that translates and contours a face of the instance to a face of the workpiece. Extend is particularly useful for a workpiece that has a nonplanar face. The result of an Extend operation is two objects, because no merging of extended faces takes place. The workpiece and the feature instance remain separate objects.

When you insert an instance of Hole_Through you must select an entry face and an exit face on the workpiece. The system can then extend the entry face of the representation to the entry face of the workpiece and the exit face of the representation to the exit face of the workpiece. The selection of the entry face and exit face makes selecting a workpiece unnecessary.

Example

The following illustration shows the result of a Thru Hole feature inserted into a model having a nonplanar face.



Merge

Merge is a local operation that extends a face of the instance to a face of the workpiece and merges the two by editing the face topology. The result of a Merge operation is one object.

The attributes EntryFace and ExitFace function for a Merge much as they do for an Extend.

Boolean

A Boolean constructs a single object from two or more operands. An operand is an object involved in a Boolean union, subtraction, or intersection. When you insert an instance of a feature using a Boolean application, the system treats that instance itself as the second operand. The workpiece you select becomes the first operand.

The particular Boolean operation (union, subtraction, or intersection) is specified within the definition of the feature. You do not have to specify it during insertion. For example, the system feature Boss_Circular is set to a Boolean union.

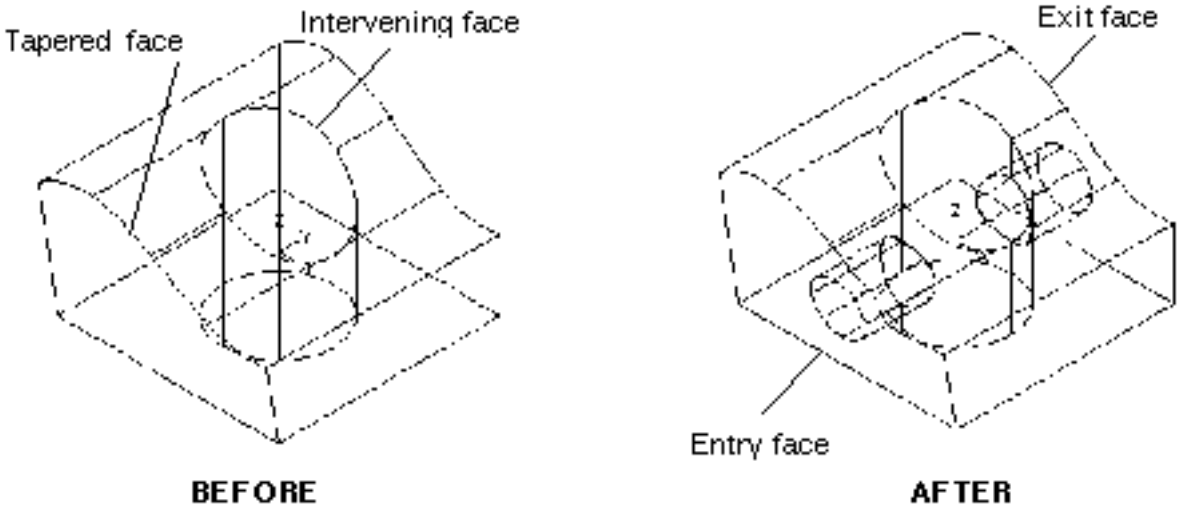
Extended Boolean

An Extended Boolean operation combines an Extend and a Boolean operation. An Extended Boolean is particularly useful for contouring the face of an instance with a nonplanar or tapered face of a workpiece that also has intervening faces. The Extend operation is performed first, then a Boolean operation to determine a solution for the intervening faces.

As is the case with a simple Boolean operation, you must select a workpiece to indicate the other operand in the operation. For an Extended Boolean you must also select an entry and exit face.

Example

The following illustration shows a case appropriate for an Extended Boolean operation, in that the workpiece has a tapered face and intervening faces.



Considerations Regarding Booleans and Local Operations

You must consider the trade-off between speed and all-inclusiveness when deciding whether to specify a Boolean, or a Merge, or an Extend option. That trade-off is as follows:

- Boolean or Extended Boolean

  The system evaluates all faces involved in a Boolean; hence the result is predictable and all-inclusive. However, if a Boolean involves many faces its execution takes more time than an Extend or a Merge.

- Merge or Extend

  The system evaluates only the faces you specify. Hence execution time is also minimal. However, you must be sure that you have specified all the necessary faces. Intervening faces between an entry face and an exit face are not detected.

In general, if you are sure of the faces that are involved in an operation, a Merge or Extend operation may be more appropriate. If you are unsure of how many faces are involved in an operation, a Boolean may be more appropriate.

If you find no advantage to choosing a particular application over the others, choose Extended Boolean. Extended Boolean is the most stable of all the applications.

NoApply

If you specify NoApply (no application), the geometry of the workpiece is not modified. The feature instance is inserted in place and is a separate object.

System Features

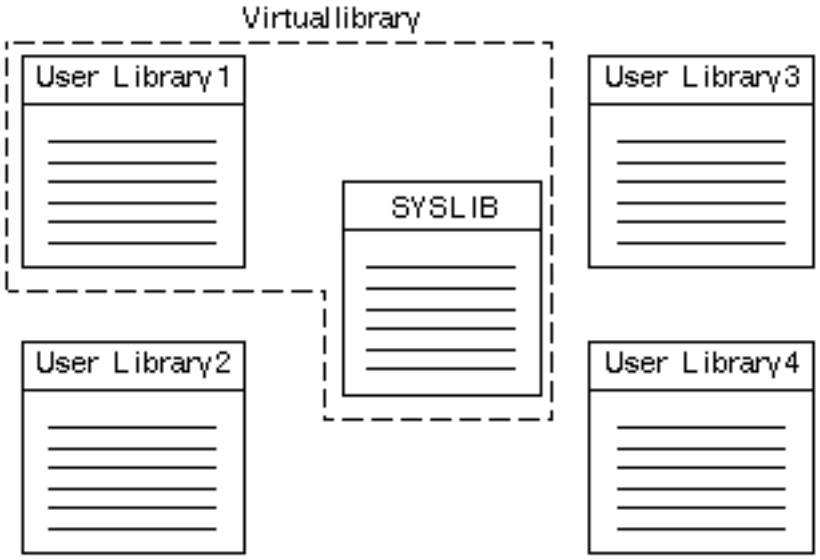
SYSLIB

The system supplies a system library, called SYSLIB, that contains all the system features. SYSLIB is always active and is read-only, hence you cannot directly modify a system feature.

The Virtual Library

You can have access to two libraries at a time: SYSLIB and a single user-defined library. Together, SYSLIB and the user library comprise a virtual library.

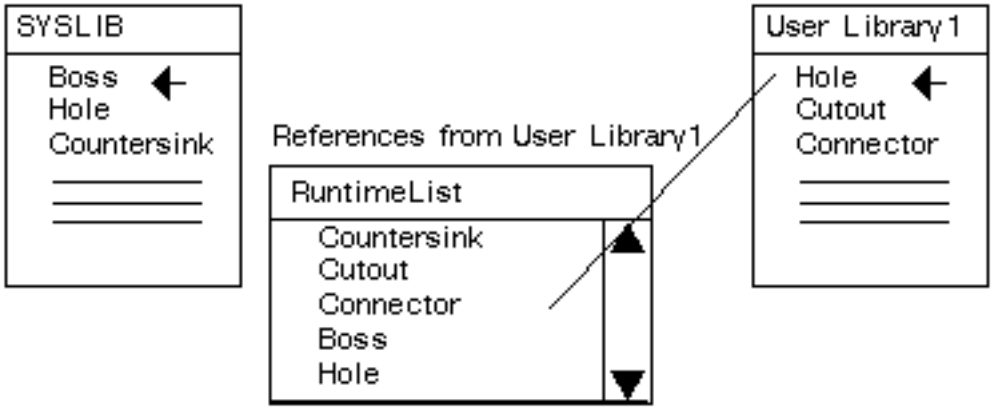
Example: The following illustration shows a virtual library comprised of the system library and User_Library1.



Same Feature Name Exists Across Libraries

If the same feature name exists across the two libraries, the feature in the user-defined library overrides the system library.

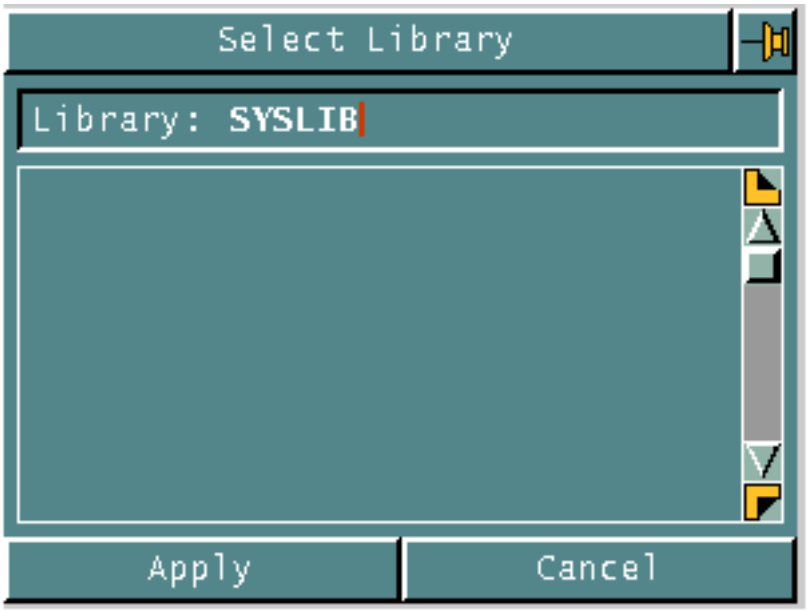
Example: If a feature called Hole exists in both the system library and the user-defined library, the system references the feature Hole in user_library1. The following illustration shows such a case.



Library Options

Library options are relevant to choosing a user-defined library.

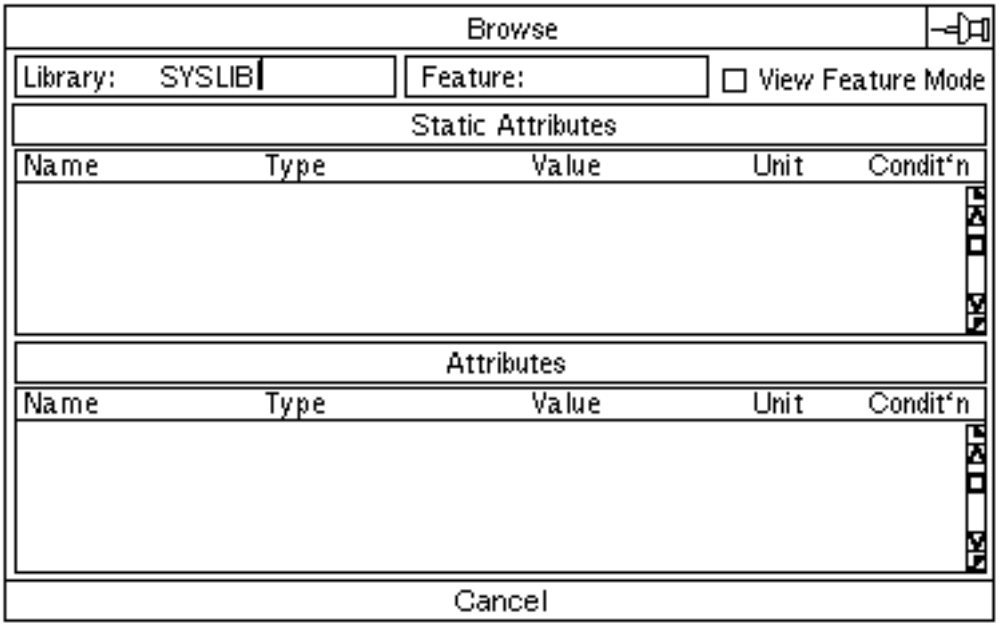
Displays the Select Library property sheet. The library you choose from that menu plus the system library comprise the virtual library.



Displays the Browse property sheet which is shown in the following illustration. The fields within that menu are as follows:

- The Library field accesses a runtime list of libraries.
- Once you make a library selection, you can use the Feature field to access a list of features within that library.
- Once you choose a feature, a list of its attributes appears.

- View Feature Mode allows you to view a feature, of the active library, before you actually insert it. View Feature Mode was discussed earlier in this chapter on page 2-4.



Setting the Tolerance Value for System Library Features

The system library features used in the Metric part now use the tolerance value of 0.05 mm instead of the previous default tolerance value that was specified in inches.

Set an appropriate Epsilon value in the configuration part (c_part) so that you can also use a flexible tolerance (Epsilon) value for the system library features in a Metric part. See *Managing CADDS 5* for details on creating a configuration part.

Please note: You cannot set a flexible tolerance value for user-defined features because these features can have specific tolerance settings that you want to preserve.

Use the following steps to set the tolerance value for the configuration part:

1. Specify the required tolerance value in the configuration part.
2. Set the value of the environment variable `PARTS_SR` to the configuration part to use the set tolerance value, in the `.caddsrc-local` file.

   For example,

```
setenv PARTS_SR /home/<user_id>/parts/config/parts
```

After you complete these steps, if you insert a system library feature in an active part, the tolerance value you specified is used.

If you do not set the value of the environment variable `PARTS_SR`, the default tolerance value of 0.05mm is used when you insert a system library feature in a Metric part.

Categories of System Features

System features can be categorized as additive or subtractive. For example, the system feature Hole is a subtractive feature because it subtracts material from a workpiece, while the system feature Boss is additive because it adds material to a workpiece. Some subtractive features can be further categorized as thru holes, cutouts, flats, grooves, and slots.

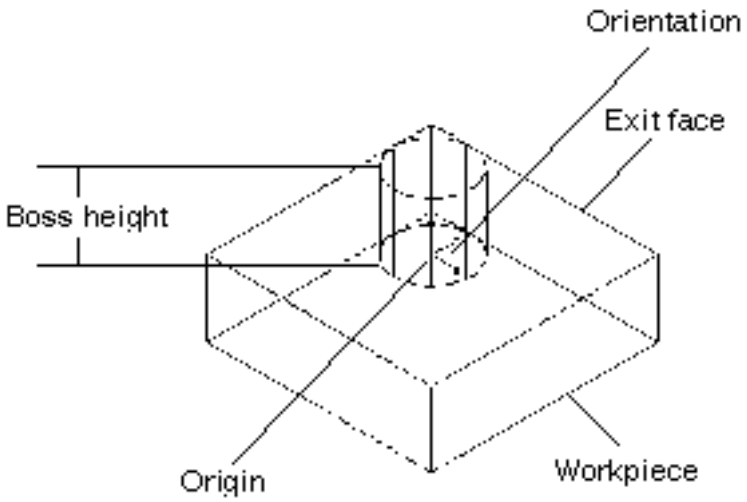
Additive Features

Additive system features are protrusions on the face of a workpiece. Note the following regarding additive features:

- The key attribute of additive features is their height above the exit face.
- An additive feature protrudes along the positive z-axis and its origin is at the base of the protrusion.
- You can apply an additive feature with all four application types (Extend, Merge, Boolean, Extended Boolean) or specify no application. If you use an Extend, a Merge, or an Extended Boolean option, you must select an exit face.

Example

The following illustration shows an instance of the additive feature Boss_Circular and its key attributes.



The following are the formal names for system features that are additive:

- Boss_Circular
- Tab_Rectangular

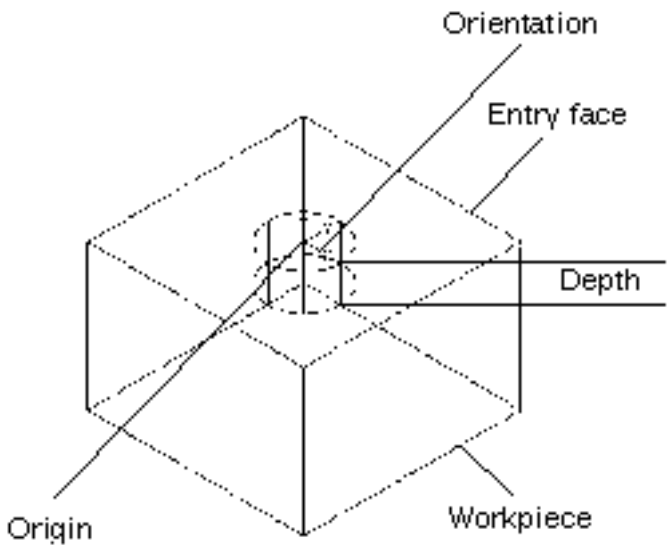
Subtractive Features

Subtractive system features are depressions in the face of a workpiece. Note the following regarding a subtractive feature:

- The key attribute is its depth into the entry face. That depth is oriented along the negative z-axis.
- The origin is on the entry face at the center of the opening.
- You can apply a subtractive feature using all four application types or specify NoApply. If you use an Extend, Merge, or Extended Boolean operation, you must specify an entry face.

Example

The following illustration shows an instance of the subtractive feature Hole_Blind and its key attributes.



The following are the formal names for system features that are subtractive:

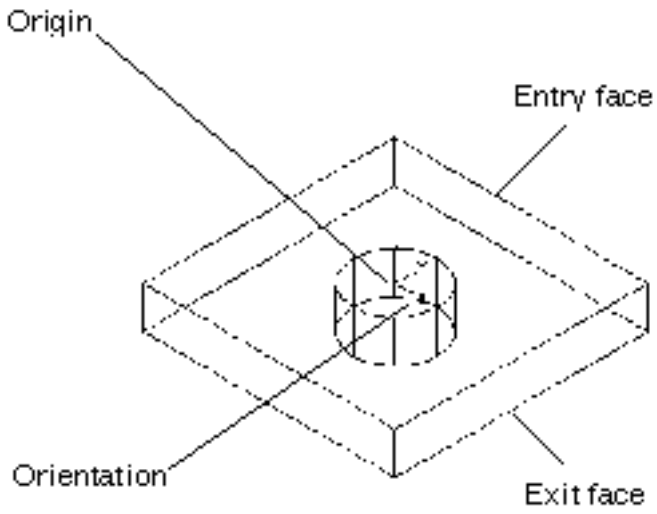
- Counterbore
- Countersink
- Hole_Blind
- Hole_Csunk
- Hole_CsunkConical
- Hole_CsunkRound
- Hole_ConicalBottom
- Hole_RoundBottom
- Hole_Cbored
- Hole_CboredConical
- Hole_CboredRound
- Pocket_Rectangular

Thru Holes and Cutouts

Thru holes and cutouts are subtractive but, more specifically, they are passages through a workpiece. Note the following regarding thru holes:

- The key attributes are the entry face and exit face and the passage is along the negative *z*-axis into the entry face.
- The origin is on the entry face, at the center of the opening.
- You can apply a thru hole or cutout with a Merge, an Extend, or Extended Boolean.
- You must select an entry face and an exit face. (Note that a Boolean and a NoApply do not utilize the entry face and exit face and thus may not produce a correct geometric representation of a passage.)

Example: The following illustration shows an instance of a thru hole and its key attributes.



The following are the formal names for system features that are passages:

- Cutout_Circular
- Cutout_Rectangular
- Hole_Through
- Hole_CsunkThru
- Hole_CboredThru

Flats and Grooves

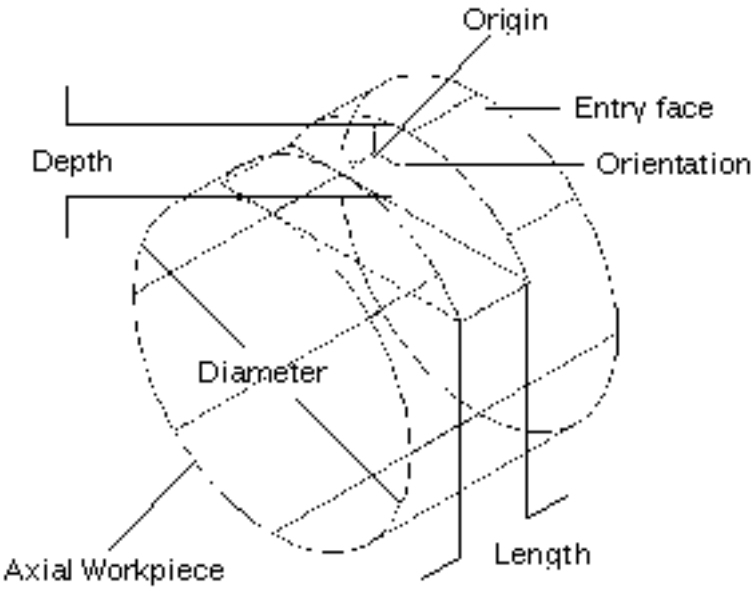
Flats and grooves are subtractive, but more specifically they are depressions on the face of an axial workpiece. The key attributes of grooves and flats are the diameter, which is actually the diameter of the axial workpiece, and the depth of the flat or groove. Note the following regarding the orientation, origin, and application of flats and grooves.

- Orientation:
  - The orientation of a flat is such that the z-axis is perpendicular to the center line of the axial workpiece and the x-axis runs along the center line of the axial workpiece.
  - The orientation of a groove is such that the z-axis runs along the center line of the axial workpiece.

- Origin:
  - The origin of a flat is the middle of the entry face at the center of the opening.
  - The origin of a groove is on the workpiece axis at the center of the groove.
- Application:
  - You can apply flats with any of the four applications.
  - You can apply a groove with a Boolean application or you can specify no application.

Example: The following illustrations shows an instance of a flat and a groove inserted into an axial workpiece.

Figure 2-8 Instance of a Flat

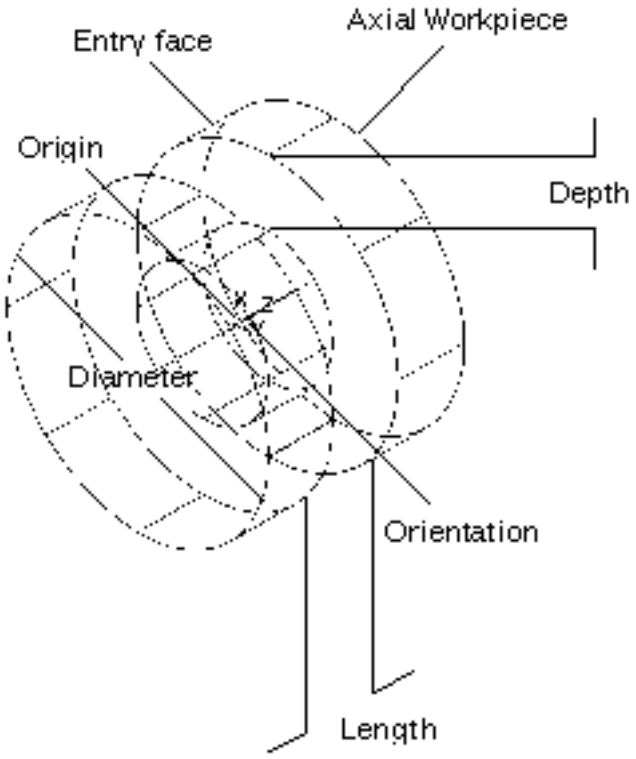


Warning

Be aware of the following restrictions when you use a Merge or Extend to insert an instance of a flat into an axial workpiece. If you cannot comply with these restrictions, use a Boolean application.

- The depth must be less than half the diameter of the workpiece.
- The length must not extend to an outer face of the workpiece.

Figure 2-9 Instance of a Groove



The following are the formal names for system features that are either flats or grooves:

- Flat
- Flat_Tapered
- Groove_Rectangular

Full and Partial Slots

Slots are depressions in a workpiece. You can insert a slot that passes either fully or partially through a workpiece. Note the following regarding the key attributes, orientation, origin, and application of full and partial slots.

- Key attribute:
  - The key attribute for a partial slot is its length.
  - The key attribute for a full slot is its exit face.

- Orientation:
  - The orientation of both slot types is such that the z-axis runs out of the top face, the face on which the slot runs through, and the y-axis runs into the entry face.
- Origin:
  - The origin of both slot types lies on the entry face at the top, center of the opening.
- Application:
  - You can apply a partial slot using any of the four application types or specify no application. Note that Merge, Extend, and Extended Boolean require the attributes EntryFace and TopFace.
  - Full slots require that you select an entry face, an exit face and a top face. Thus for full slots you can only select a Merge, Extend, or Extended Boolean. (Note that a Boolean and a NoApply do not utilize the entry face, top face, and exit face and thus may not produce a correct geometric representation of a full slot.)

Example: The following illustrations show an instance of a full slot and an instance of a partial slot inserted into the same workpiece.

Figure 2-10 Full Slot

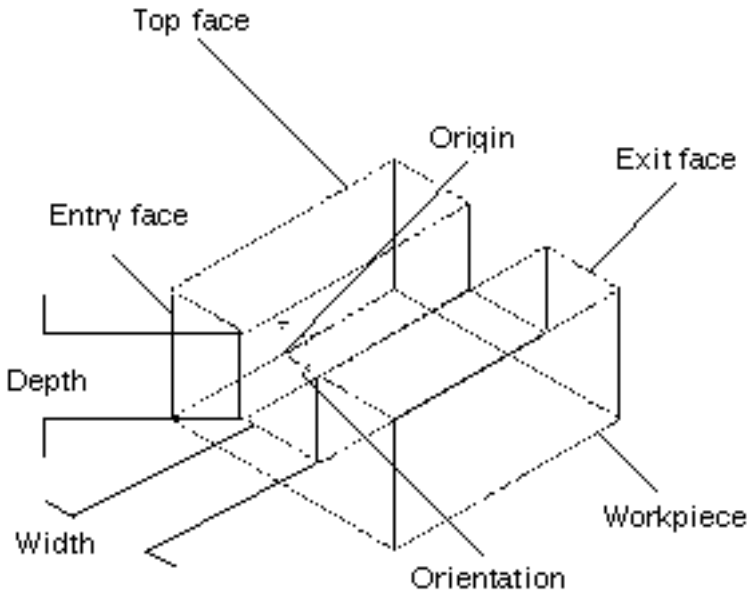
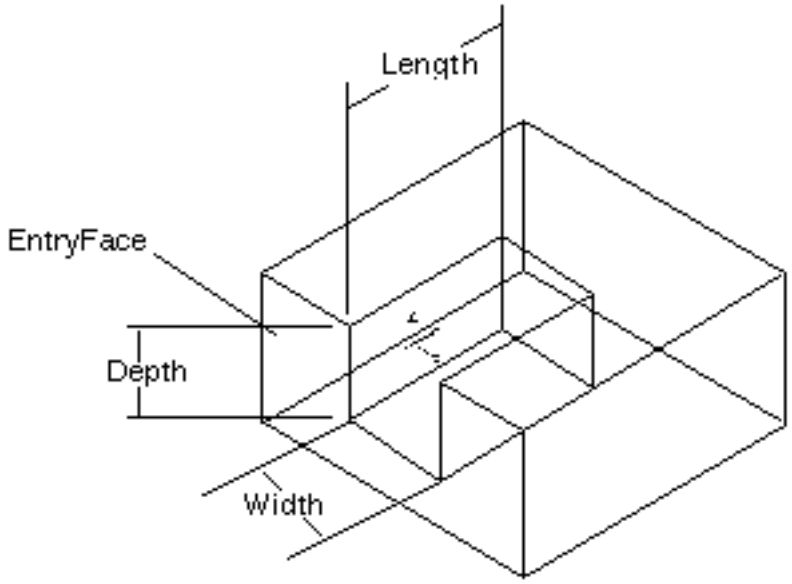


Figure 2-11 Partial Slot



The following are the formal names for system features that are either full or partial slots:

- Slot_FullRectangular
- Slot_PartialRectangular

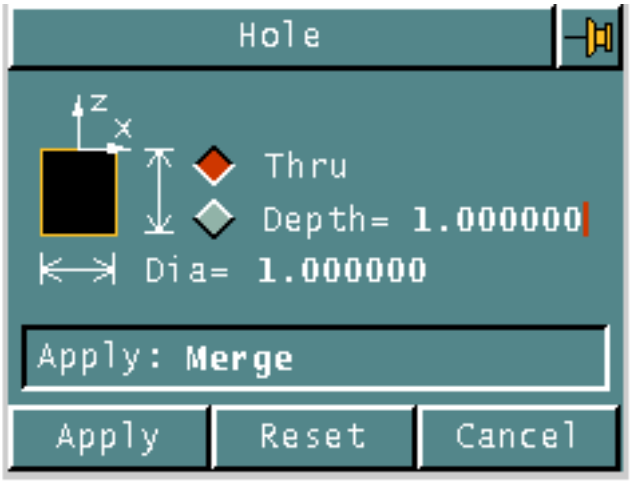
System Feature Options from the Features Task Set

The following lists the system feature options on the Features task set.

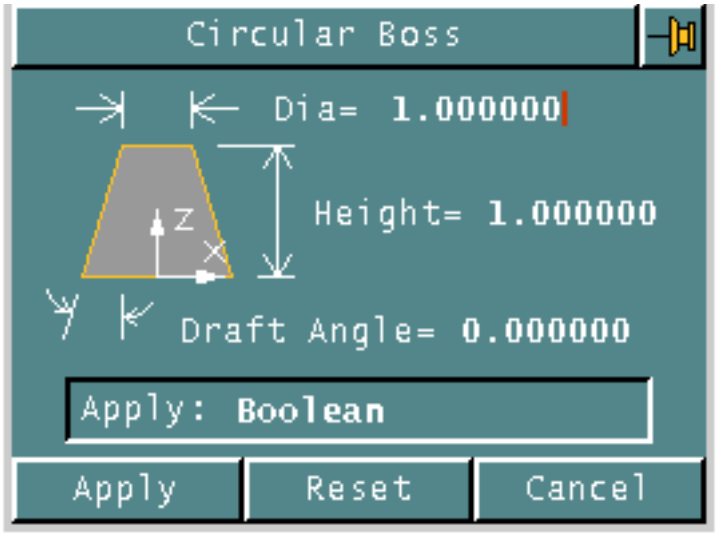
Options in red: Some options within the Features menu display menus with options highlighted in red. You must choose and provide a value for those highlighted options.

Description of Options

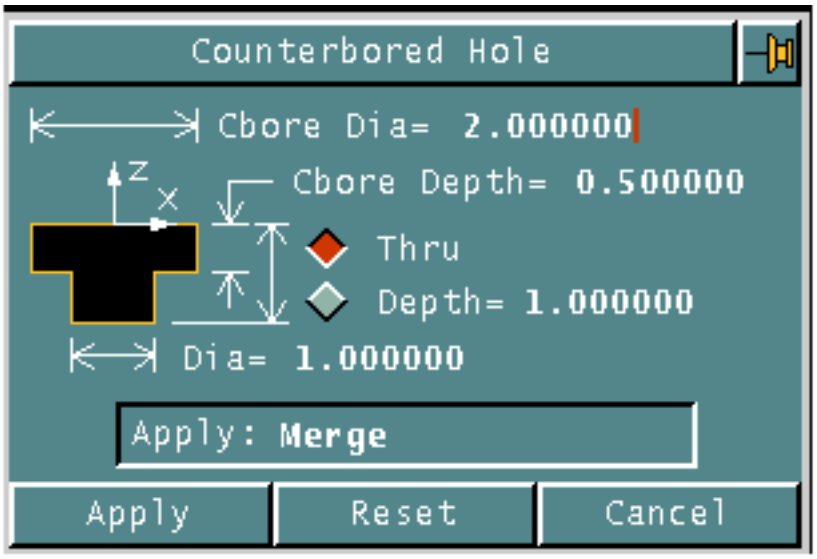
Displays the Hole property sheet.



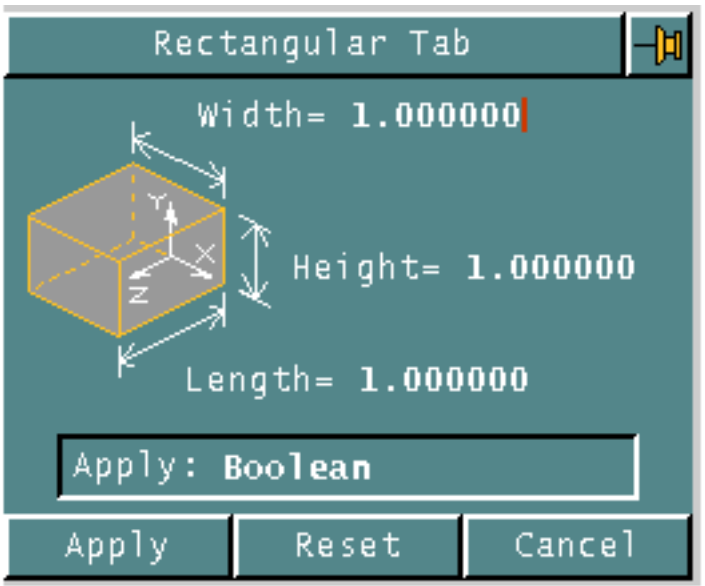
Displays the Circular Boss property sheet.



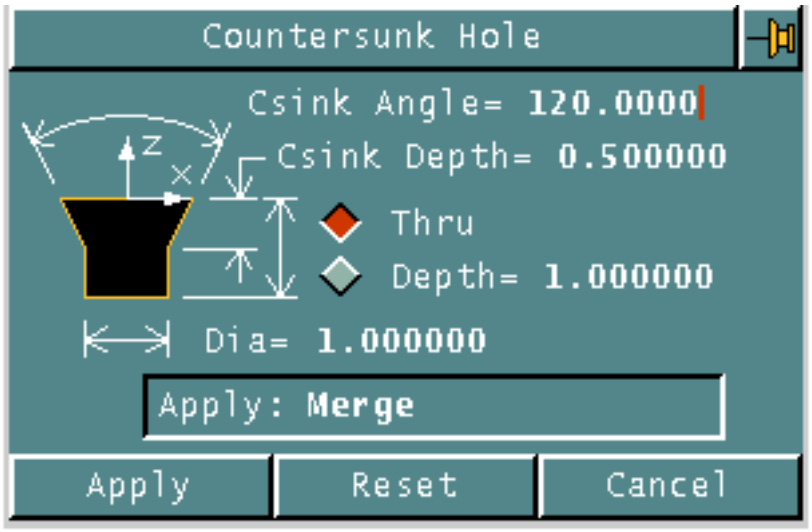
Displays the Counterbored Hole property sheet.



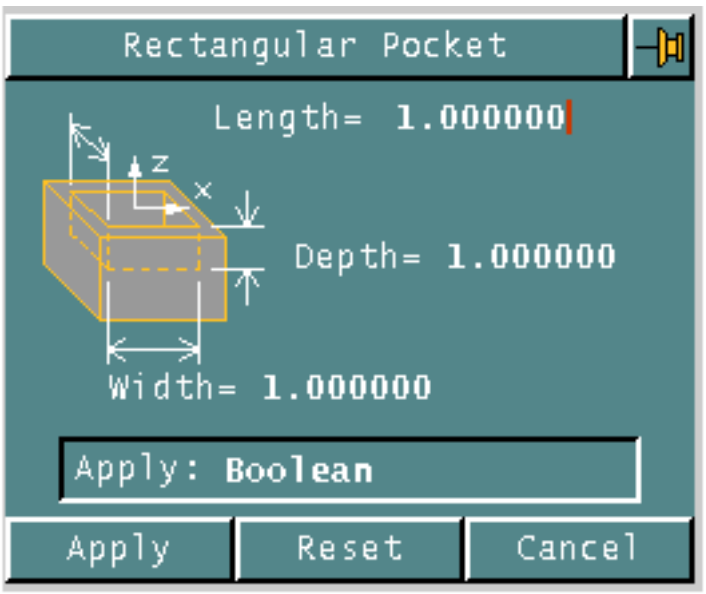
Displays the Rectangular Tab property sheet.



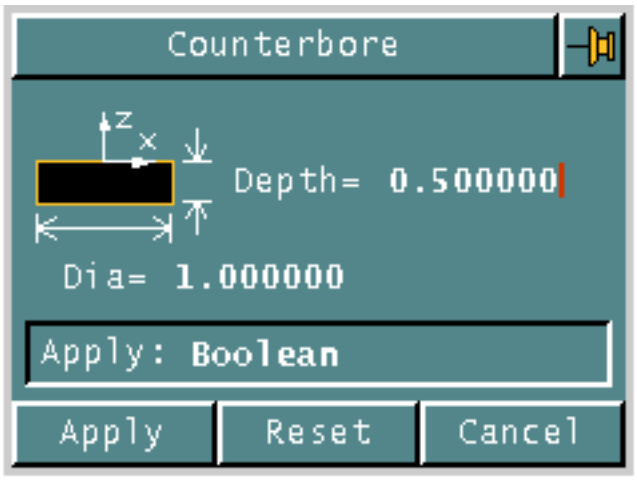
Displays the Countersunk Hole property sheet.



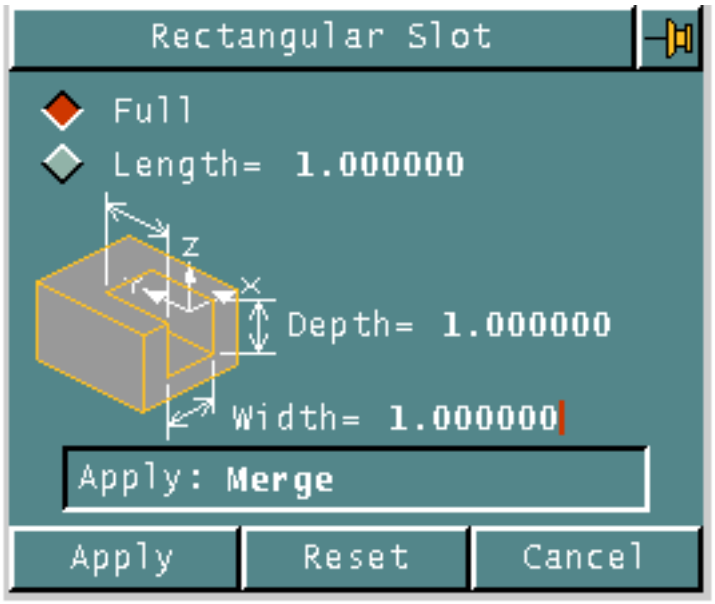
Displays the Rectangular Pocket property sheet.



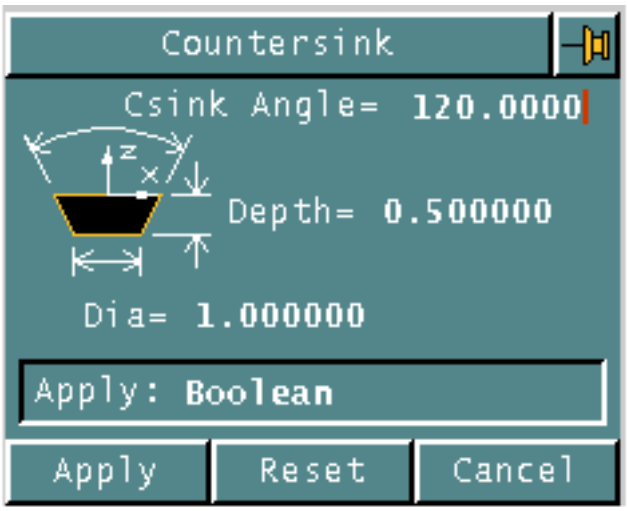
Displays the Counterbore property sheet.



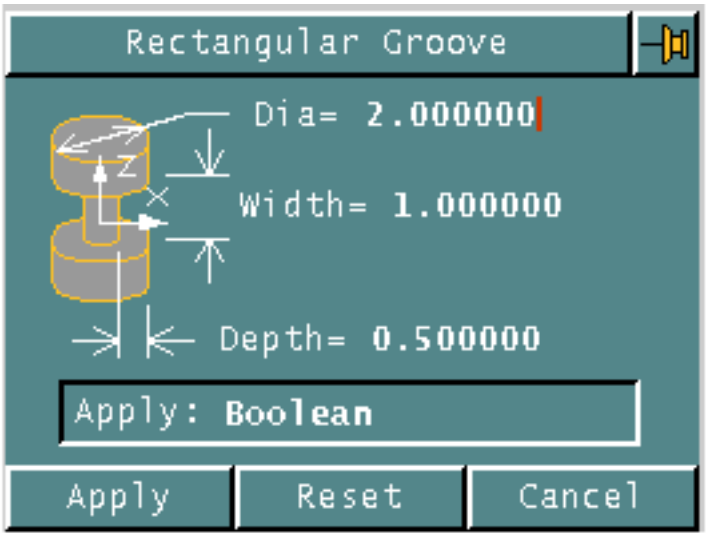
Displays the Rectangular Slot property sheet.



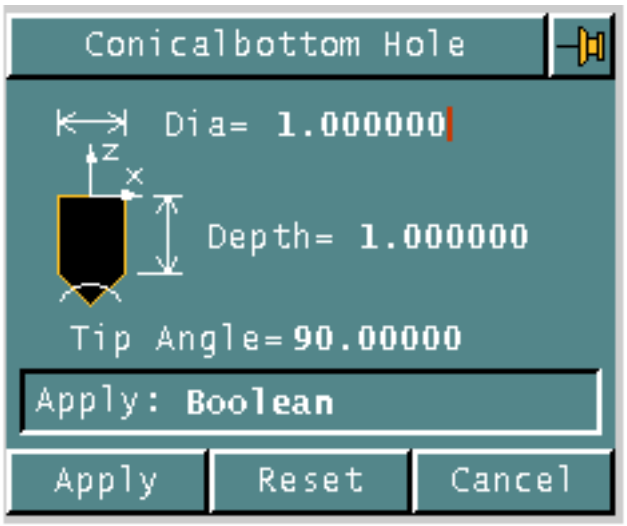
Displays the Countersink property sheet.



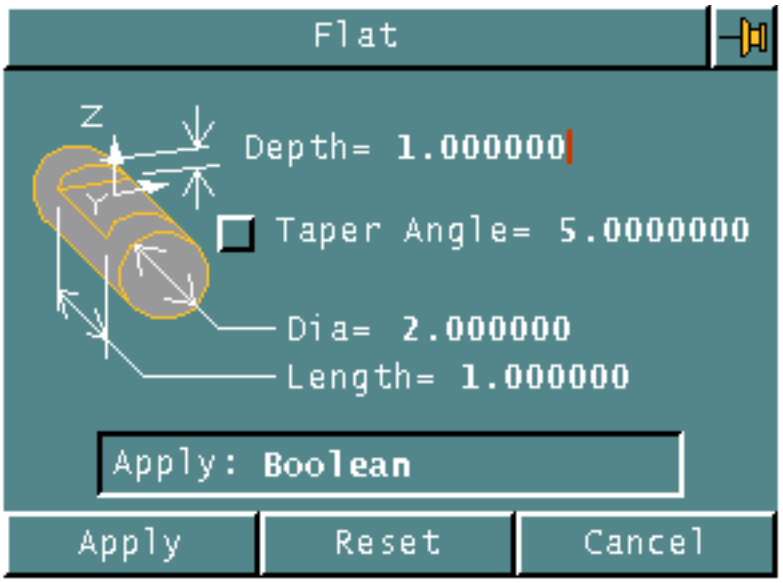
Displays the Rectangular Groove property sheet.



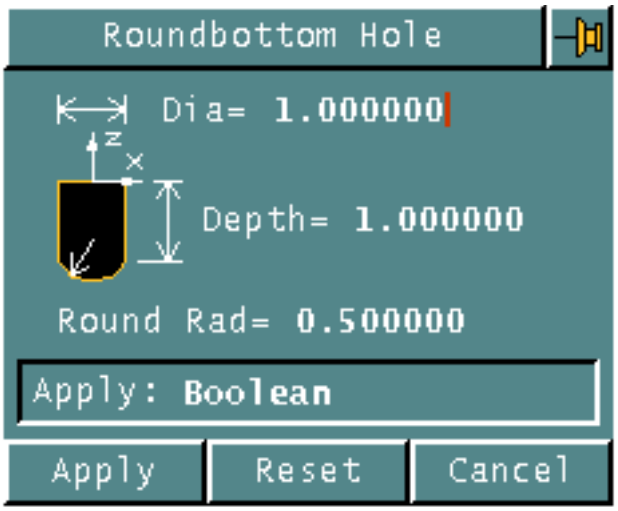
Displays the Conicalbottom Hole property sheet.



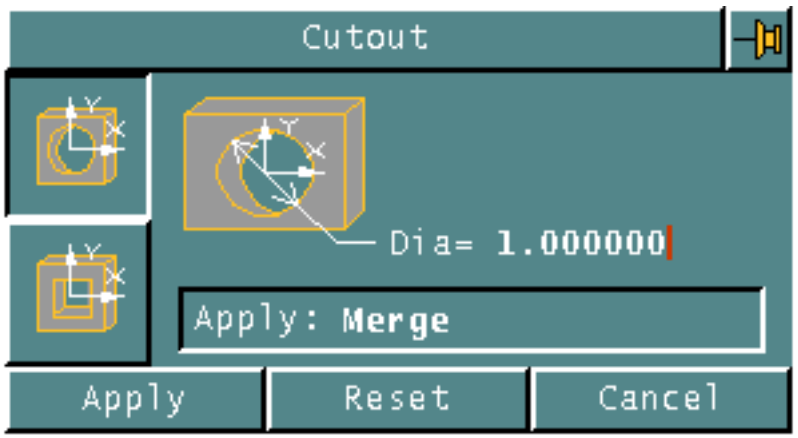
Displays the Flat property sheet.



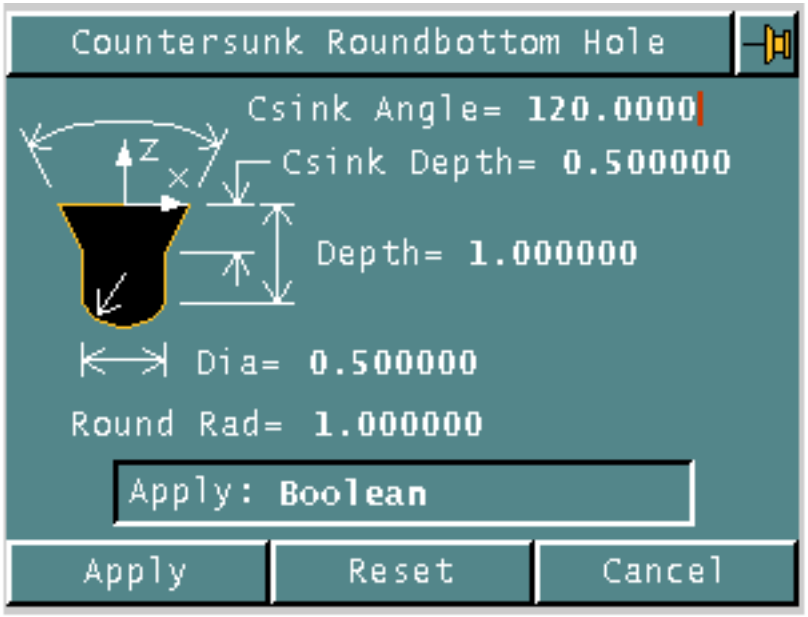
Displays the Roundbottom Hole property sheet.



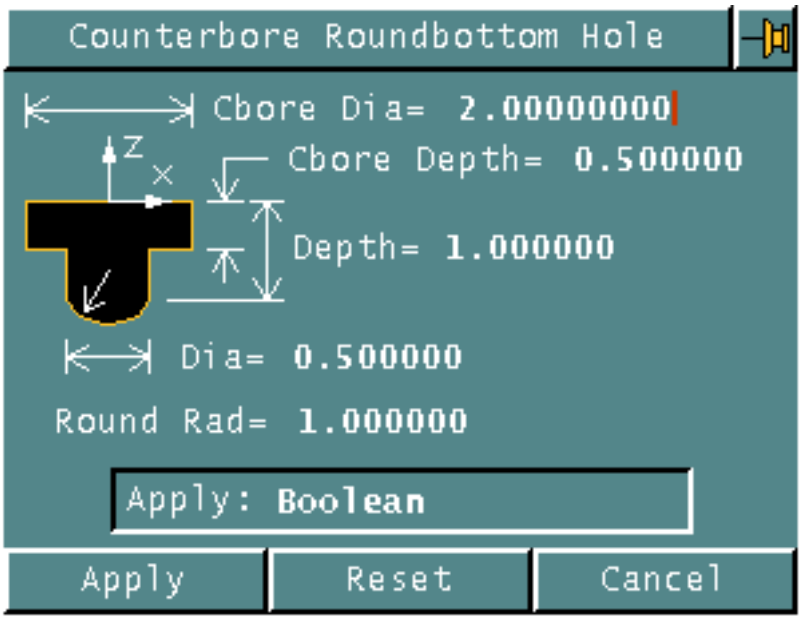
Displays the Cutout property sheet.



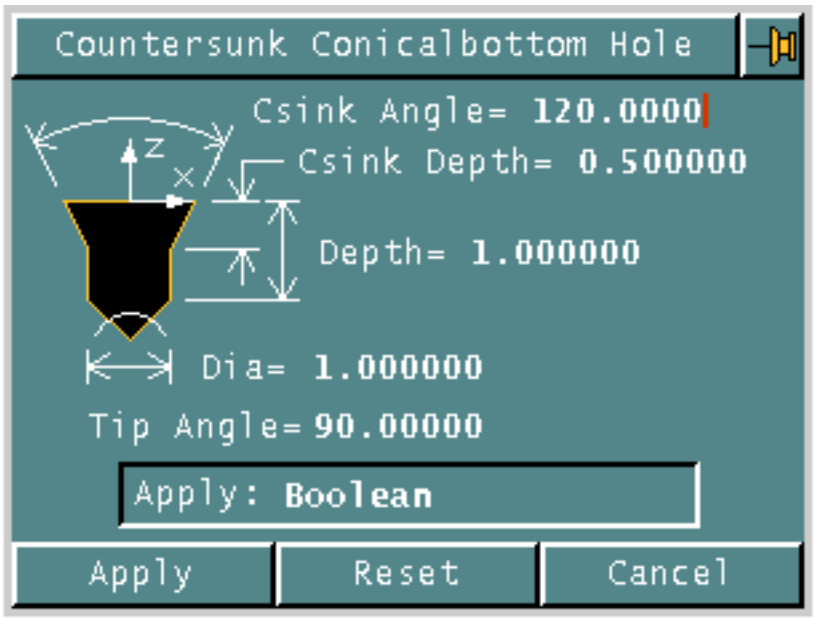
Displays the Countersunk Roundbottom Hole property sheet.



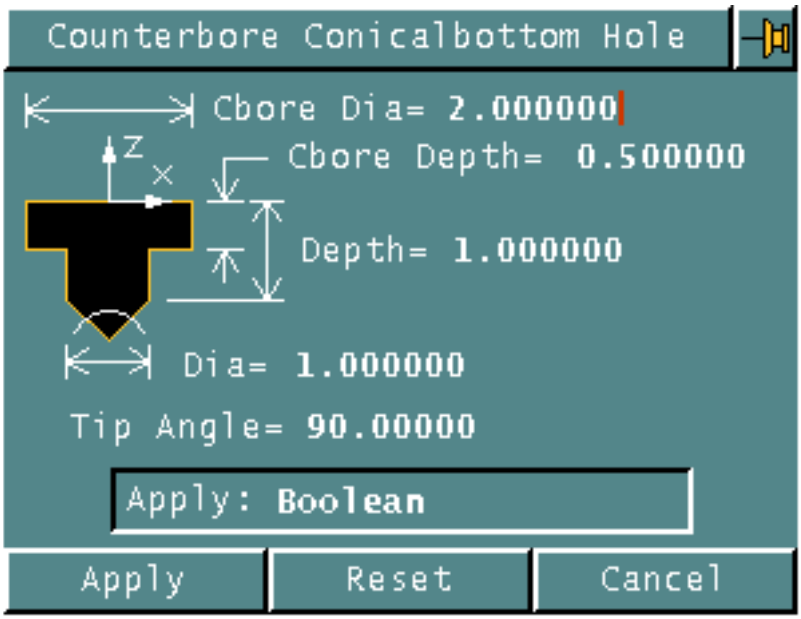
Displays the Counterbore Roundbottom Hole property sheet.



Displays the Countersunk Conicalbottom Hole property sheet.

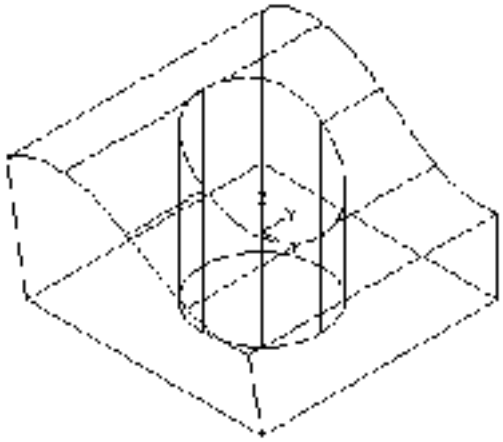


Displays the Counterbore Conicalbottom Hole property sheet.



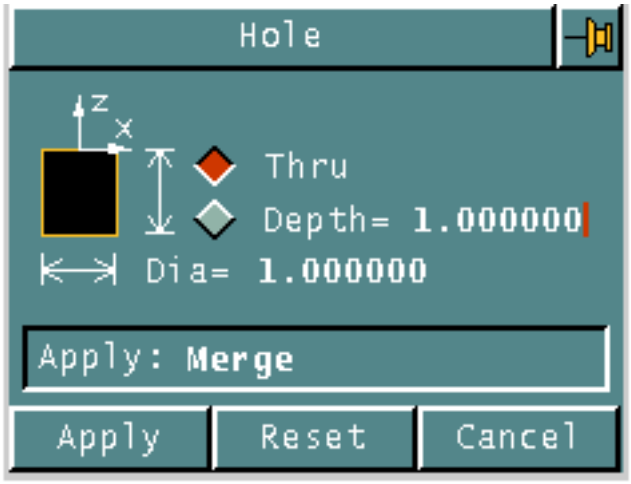
Sample Session: Inserting an Instance

The following procedure describes how the instance of the system feature Hole_Through was inserted into the solid shown in the illustration below.



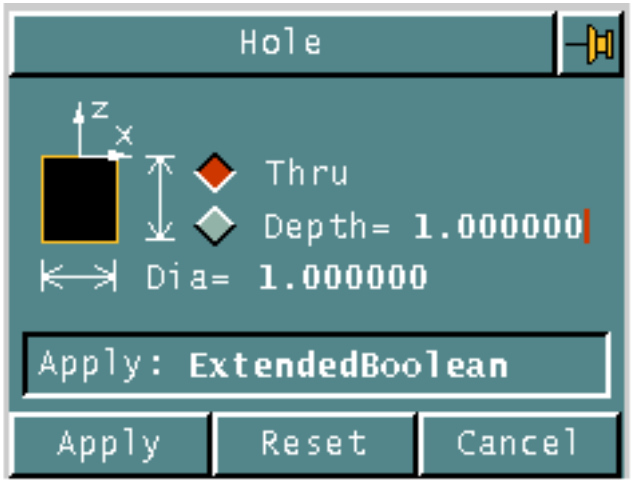
1. Choose the Hole option on the Features task set.

   The property sheet shown appears as shown below.



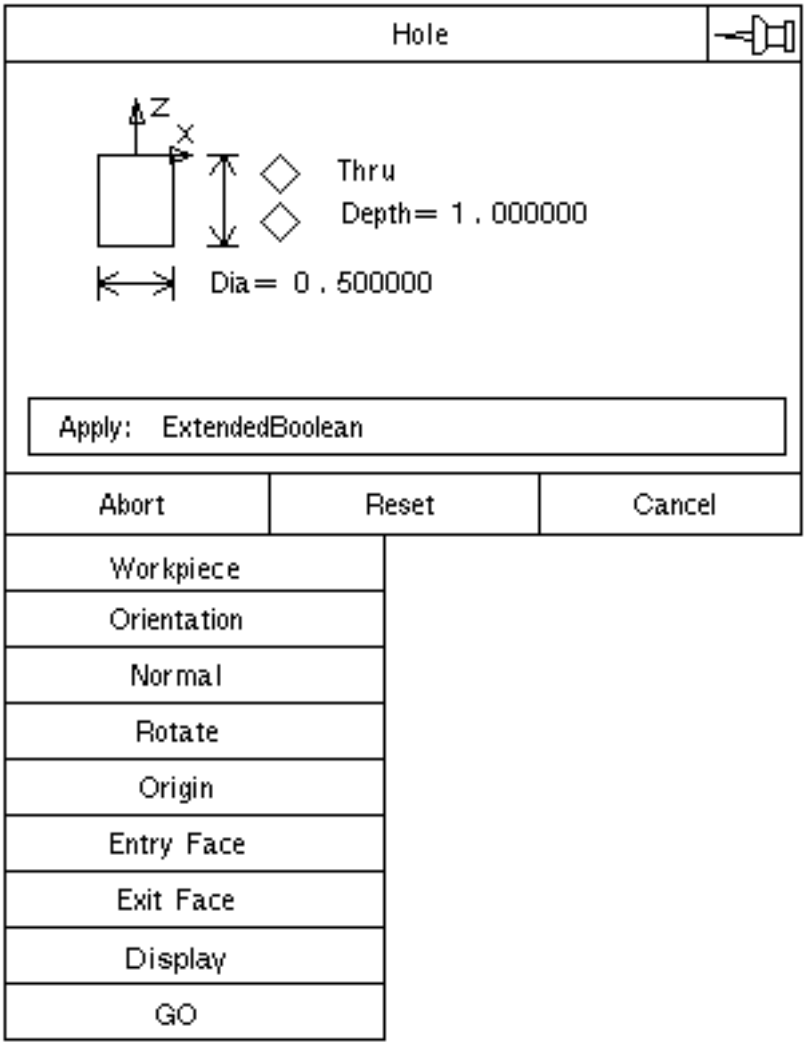
2. Choose Thru, change the diameter to 0.5, and change the application to Extended Boolean.

The property sheet updates accordingly.



3. Choose Apply.

A set of attributes is displayed for which you can assign values.



4. Choose Workpiece.

   You have specified the first operand for the Boolean subtraction. The instance(s) is the other operand.

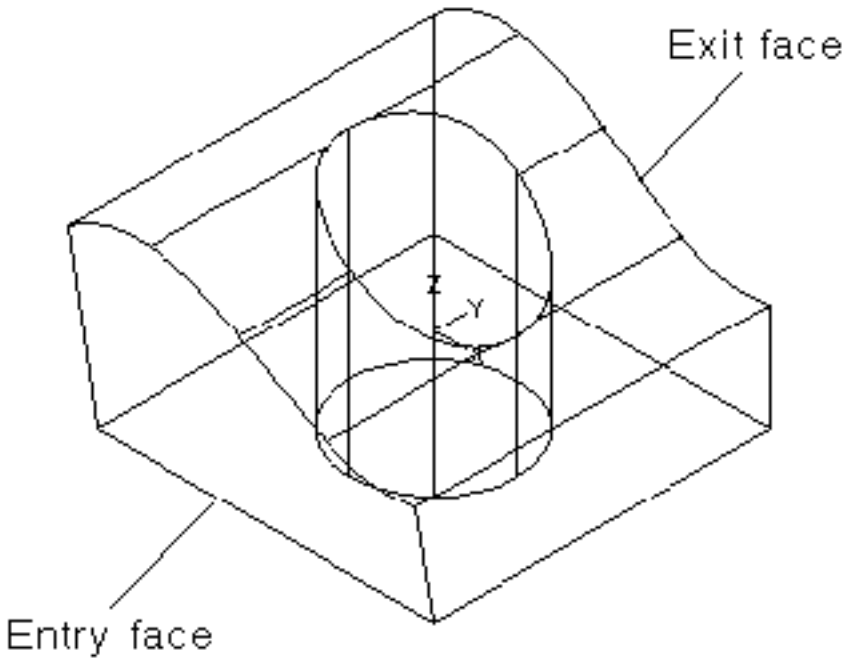
5. Choose Display from the pulldown menu that appears to preview the default orientation of the inserted feature relative to the workpiece.

6. Reorient the instance by using the Orientation option and selecting the Cplane called FRONT.

The instance is now inserted with an orientation relative to the Cplane FRONT.

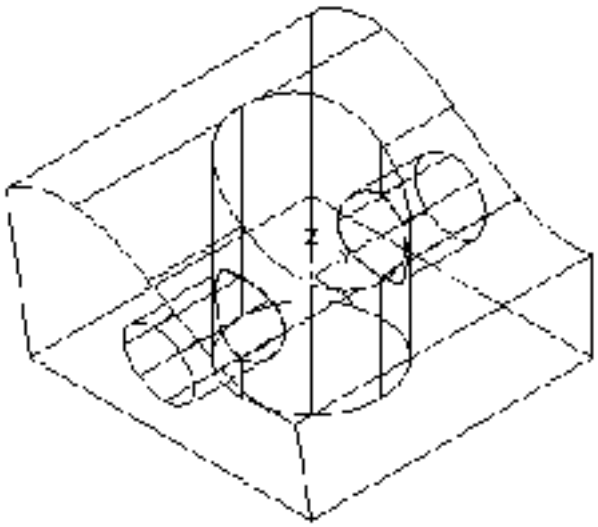
7. Select an entry face and an exit face.

   The system extends the instance entry face to the entry face you just selected and likewise for the exit face using the orientation you specified.



8. Choose Go on the pulldown menu.

The system inserts an instance. Since no origin was specified, the system uses the default, which is the [0,0,0] of the selected orientation. The following illustration shows the inserted instance.

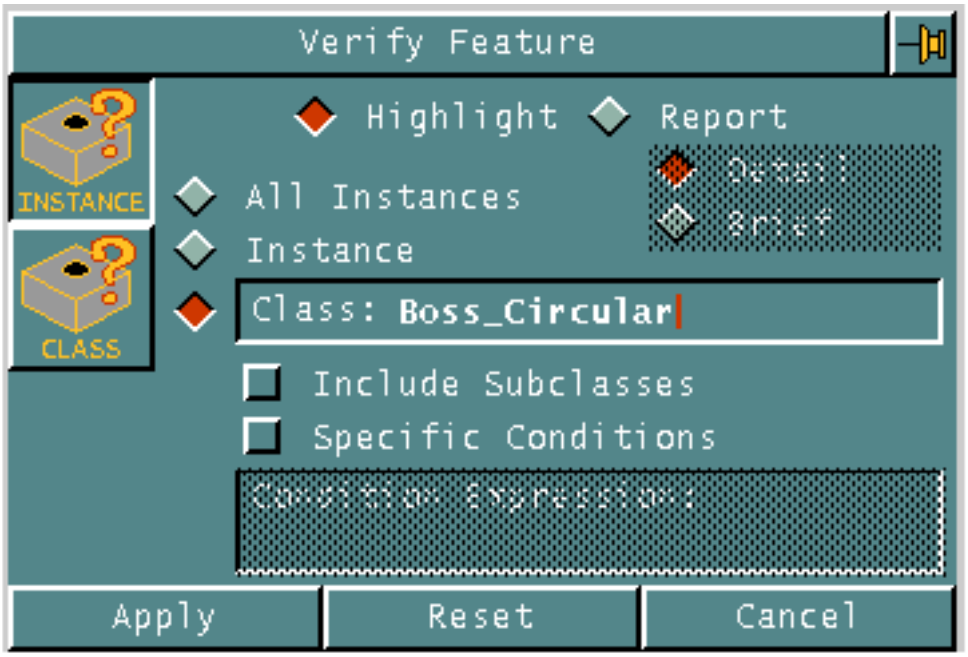


Instance Information Retrieval: The Verify Feature Option

The Verify Feature option on the Features task set displays a property sheet that provides two options for verifying features. One option, Instance, graphically highlights or displays a report about the feature instances within the current part that meet the conditions you have set. The other option, Class, displays a report about instances of the feature definition (class) you choose.

When Instance is Active

By default Instance is active. When it is active, the Verify Feature property sheet appears as shown in the following illustration. You can use that property sheet to either display a report or highlight instances that meet the conditions you have set.



The following describes the options when Instance is active:

- Highlight highlights the instances that meet the conditions you have set.
- Report displays a report on instances that meet the conditions you have set.
- Instance reports upon the instance you choose.
- All Instances highlights or reports upon all feature instances in the part.

- Class displays a runtime list of feature definitions (classes) within the part. Choose one and choose Apply to highlight or report on all instances of the class you chose. If no class is chosen, all instances are highlighted or reported upon.
- Include Subclasses works with Class to include instances that are subclasses of the class you chose.
- Specific Conditions activates the Condition Expression box. You can enter a conditional expression directly into the box or you can choose the box to activate the Condition Expression menu.

Entering Conditional Expressions

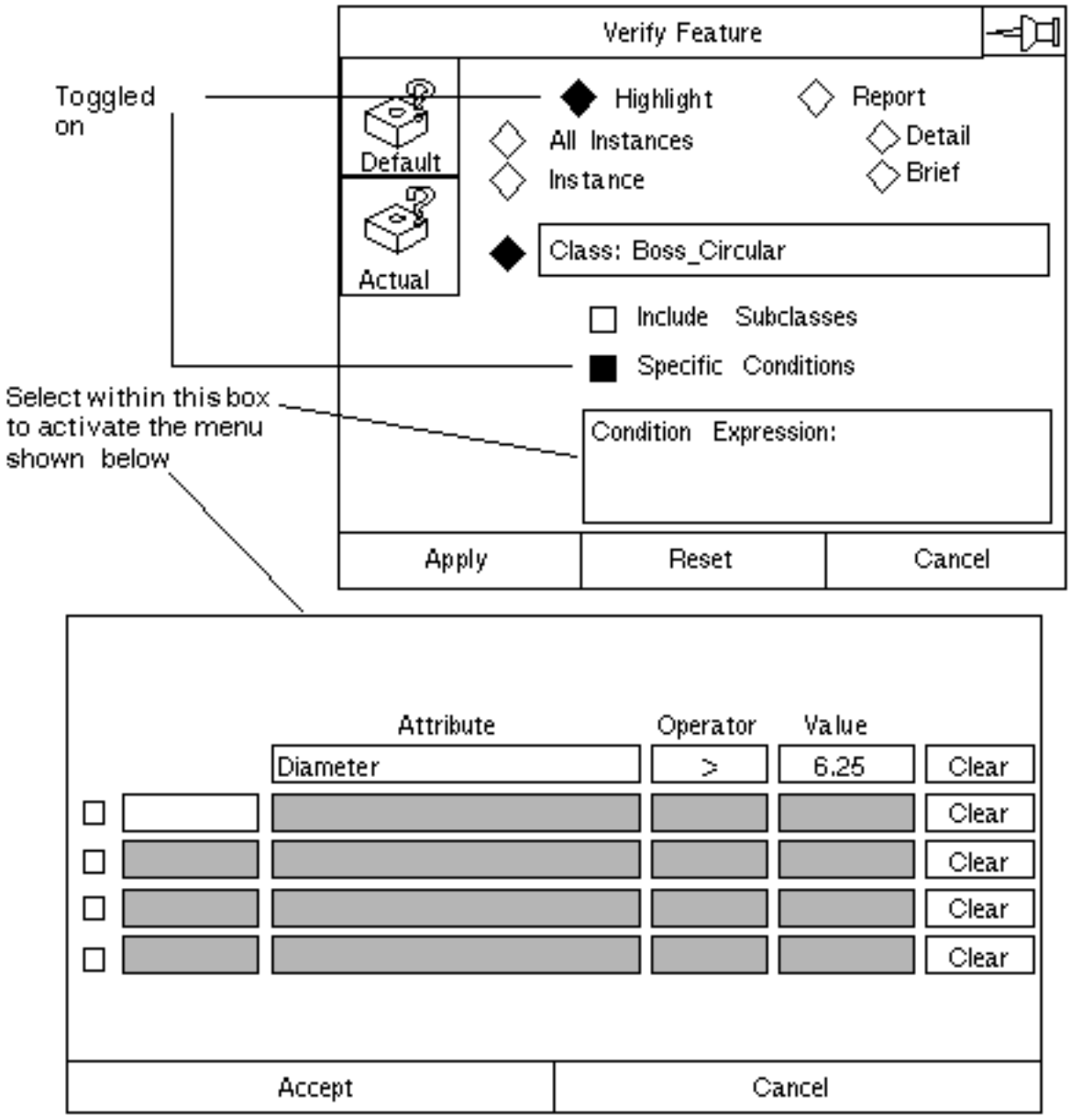
You can use any of the following operators when entering a conditional expression:

- =
- >
- <
- <=
- >=

In addition you can use the following conditional operators:

- And
- Or
- Not

The following illustration shows how you might use the Condition Expression menu to highlight all instances of Boss_Circular that have an attribute called Diameter with a value greater than 6.25.

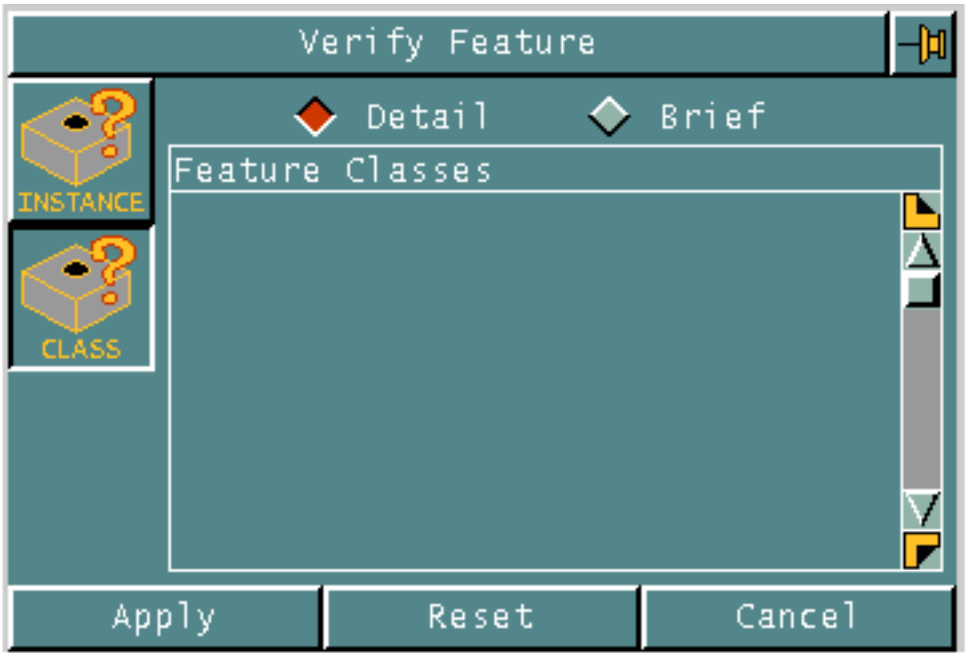


The following describes the options within the Condition Expression menu:

- Attribute displays a list of attributes of size from which you can choose.
- Operator displays a list of operators (=, <, >, and so on) from which you can choose.
- Value displays a calculator that you can use to enter a value.
- The rectangular box below displays a list of the conditional operators (and, or, not) you can choose to include in your statement. For example, you might construct the following statement: Diameter > 6.25 and Depth < 3.

When Class is Active

When Class is active, the Verify Feature property sheet appears as shown in the following illustration. You can use that property sheet to display a report about instances of the feature definition (class) you choose.



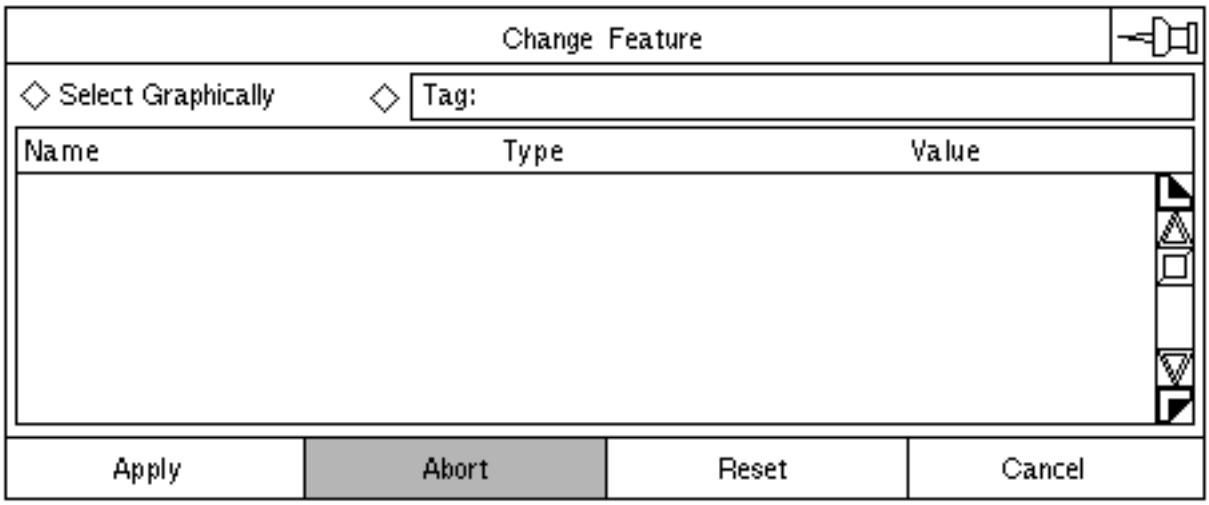
- Feature Classes displays a runtime list of feature definitions (classes) within the virtual library. Choose one and then choose Apply to display a report on the selected feature definition.
- Detail and Brief specify the amount of information to be retrieved. Detail includes information about a feature's superclasses and subclasses while Brief does not.

Instance Modification: The Change Feature Option

You can modify an instance by changing its attribute values.

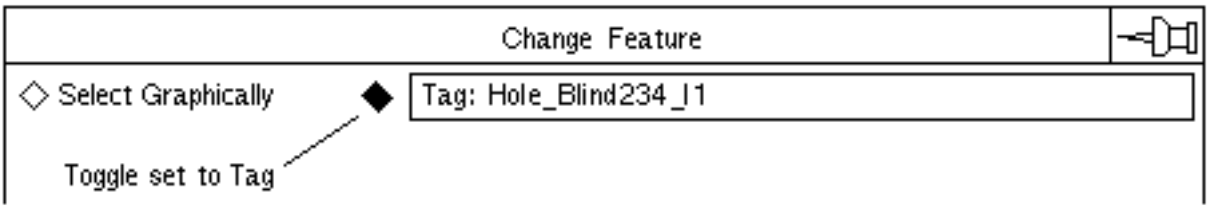
The Change Feature option allows you to change the attribute values of an instance. You can:

- Select an instance graphically or by its tag
- Choose an attribute from a list and modify its value
- Choose a parametric dimension of that instance and change its value

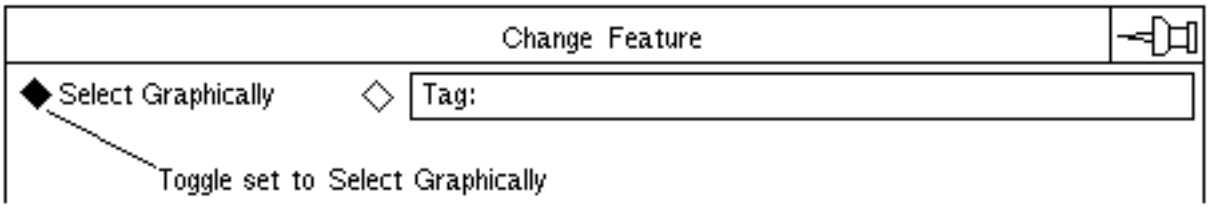


Selecting an Instance

If you want to choose an instance by name, activate the Tag toggle and enter the tag of the instance you want as shown in the following example. Use the Verify Feature option on the Features task set to display a report that includes instance tags.

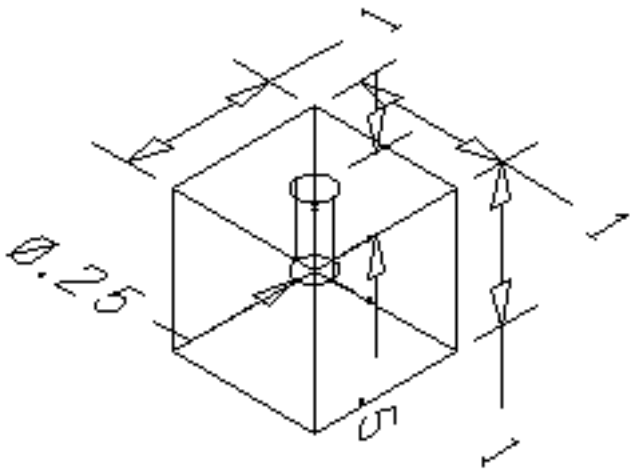


If you want to select an instance graphically, choose Select Graphically and select an instance.

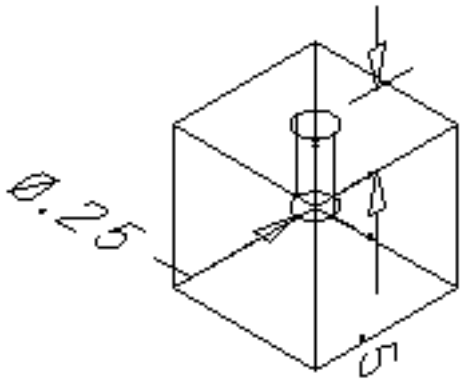


Once you select an instance, Change Feature blanks all parametric dimensions not associated with that instance and unblanks all parametric associated dimensions. Such blanking allows you to easily recognize only those parameters that are associated with the instance.

The following illustration shows an instance of Hole_Blind inserted into a block.



The next illustration shows the instance after it has been selected while using the Change Feature option. Notice all parameters have been blanked except those associated to the Hole_Blind instance.



Choosing and Modifying Attributes

The Change Feature option allows you to change the geometric and non geometric attributes of an instance. However, only attributes of the following types can be changed:

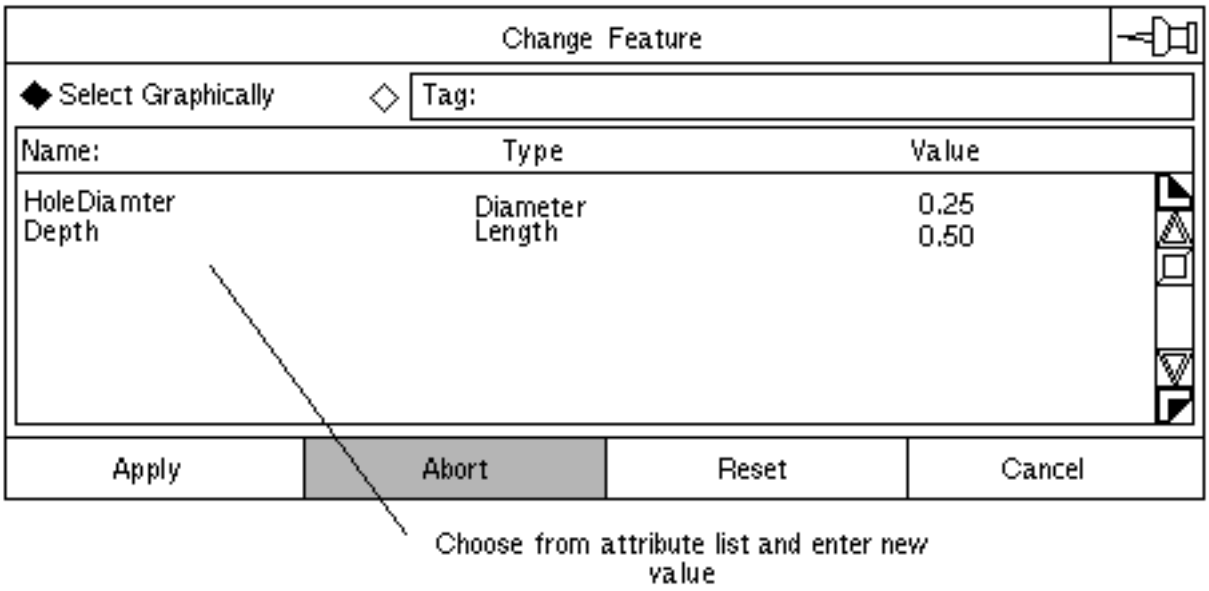
- Integer
- Cardinal
- Real
- Length
- Radius
- Diameter
- Angle
- HalfAngle
- String
- Name

Warning

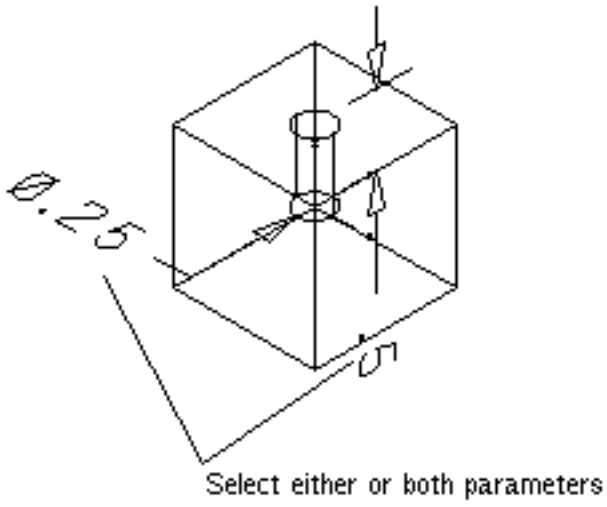
You cannot use the Change Feature option on the following attribute types:

- Location
- Vector
- Cplane
- Entity
- Face
- Edge
- Profile
- Instance
- Boolean_Op
- Applytype

You can choose and modify one or more attributes of a selected instance from the Change Feature property sheet. Choose the attribute(s) you want and enter the new value(s) using the calculator that appears. The following illustration shows the Change Feature property sheet while the instance of Hole_Blind is selected.

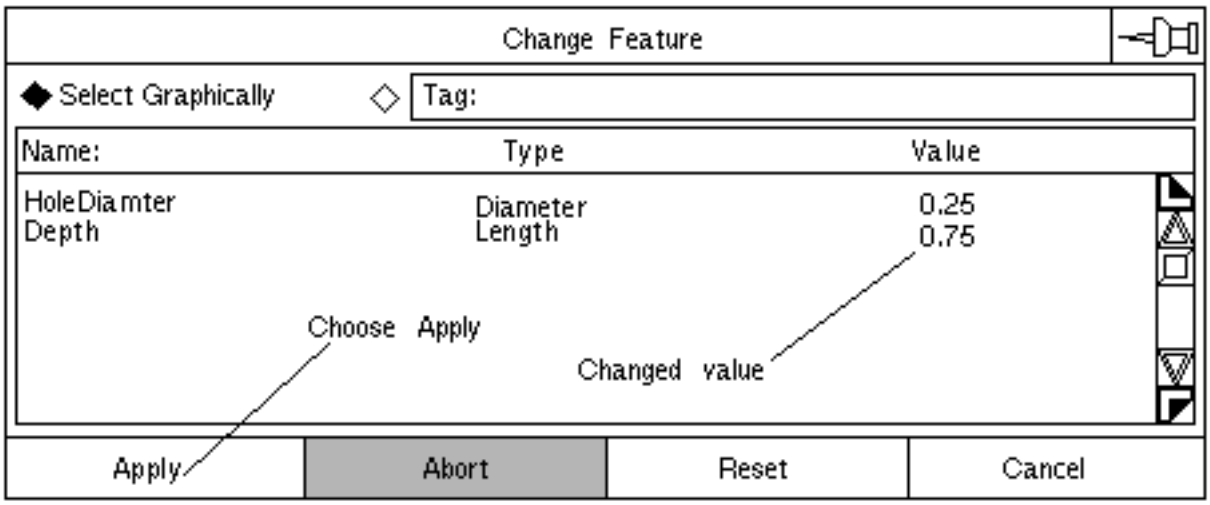


Alternately, you can choose and modify one or more attributes of a selected instance by specifying parametric dimension(s) and entering a new value using the calculator that appears. The following illustration shows the Hole_Blind instance in a selected state and indicates that you can select either or both the depth or diameter parameter of the instance. All other parameters are blanked.

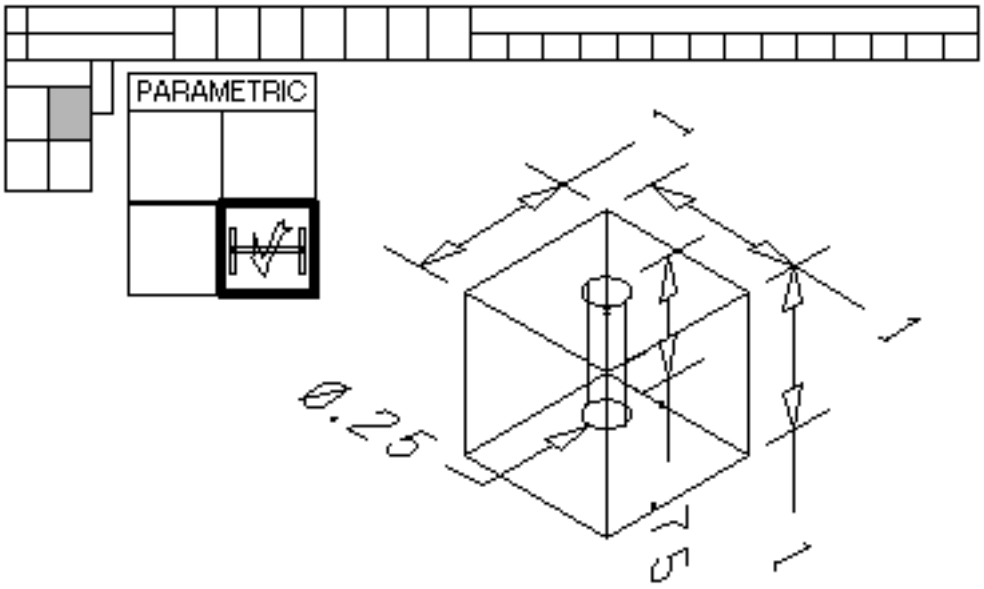


Whether you choose and modify an attribute from the list or by its parameter, the Change Attribute option responds by Highlighting the parameter attribute. (Red is the default highlighting color for changed parameters). You must then

1. Choose Apply. The following illustration shows the Hole_Blind instance with its Depth attribute changed from its initial value of 0.50 to 0.75.



2. Choose the Regenerate Model option. The system responds by regenerating the instance according to the modified values. In the case shown, the depth becomes .75. Notice how all parameters are again visible.



Chapter 3

Creating a Feature

This chapter explains how to create a feature.

- Overview of User-defined Features
- Creating a Definition
- Creating a Representation
- Sample Session: Creating a Feature

Overview of User-defined Features

You can create a library of user-defined features and insert an instance of one much like you would for a system feature.

The following steps provide a generalized outline of the tasks required to create a user-defined feature:

1. Create a definition with the Define Feature option on the Features task set.
2. Create a representation by conducting a Teach Feature session. Later in the same Teach Feature session you can associate particular attributes from the definition you specify to particular entities, parametric dimensions, parametric labels, or user dimensions.

You do not have to approach these tasks in a completely linear fashion. You can, for example, create consecutive definitions and then create a representation part for each respective definition.

Please note: You must create a definition before creating a representation for that definition.

You can use the sketches created in the Sketcher environment to create user defined features as part of the feature libraries. Using Sketcher in Feature's "Teach" mode, you can create 2D shapes which can then be used to create 3D geometry as part of feature definition.

Modifying a Definition

You can modify an existing definition. However, under certain conditions, the system locks that definition’s representation. Locking prevents you from inserting an instance of the representation.

The system locks the representation of a definition if you make any of the following modifications:

- Delete a superclass
- Delete an attribute of size or a face attribute
- Change an existing attribute’s data type

Under those conditions the system locks the modified definition’s representation along with the representations of any of its subclasses (any of its descendant subclasses).

Modifying a Representation

You can modify an existing representation. Existing instances of that feature are not affected by the modifications. Subsequent instances of that feature reflects the modifications.

Creating a Definition

Creating a definition allows you to specify a set of attributes customized to your design needs. You can utilize a system feature definition as a superclass or choose an existing user-defined feature definition.

Overview of the Process

You can create a definition within the Local Data Manager or while a part is active. For each definition you create, you must complete the following tasks:

1. Choose the Define Feature option from the Features task set.
2. Choose a user library. The definition you create is saved in that library. By default it is created in the current library.
3. Enter a definition name. If the feature is new, specify a unique name. A feature name can have a maximum of 29 characters and its first character must be a letter.
4. (Optional) Choose a superclass(es). If you specify a superclass, the definition inherits all the attributes of that superclass. By default, your definition is a subclass of the system class Feature. You can, however, change that default.
5. Enter attributes.

Entering Attributes

Entering an attribute requires that you specify the following items:

- The name of the attribute. This item is a unique string, within a definition, that identifies the attribute.
- The type. This item specifies a data type (integer, string, and so on) so that only a valid value can be assigned to the attribute. For example, if you define an attribute called thickness and identify it as the type integer, then attempt to assign to it a non integer value, that value is not accepted.
- The default value of an attribute. This item assigns a default value to an attribute in accordance with the following conditions:
  - Attributes of size are typically assigned a value during definition.
  - Static attributes cannot have their values changed when you insert an instance.
  - Dynamic attributes have a value that is simply a default that you can change when you insert an instance.

- Attributes that require a graphical selection cannot have a default value. For example, you cannot assign a default to the attribute workpiece as it requires a graphical selection when you insert an instance.

- The units. This item specifies the units used by an attribute of size.
- The material condition. This item indicates how increasing the value of an attribute of size affects the amount of material in the workpiece once you insert an instance of the feature. That effect can be any of the following:
  - Less material, a negative (-) condition
  - More material, a positive (+) condition
  - Same amount of material, an equivalent (=) condition

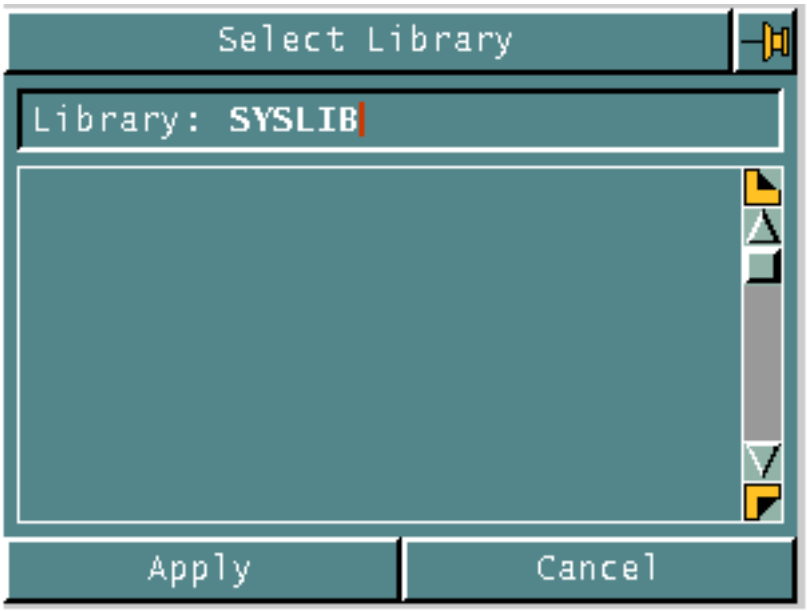
Please note: The system does not allow you to mix properties and features within the same library. Properties are discussed in Chapter 4, “Using Properties.”

Options Related to Creating a Definition

The following options are relevant to creating a definition and are on the Features task set.

Description of Options

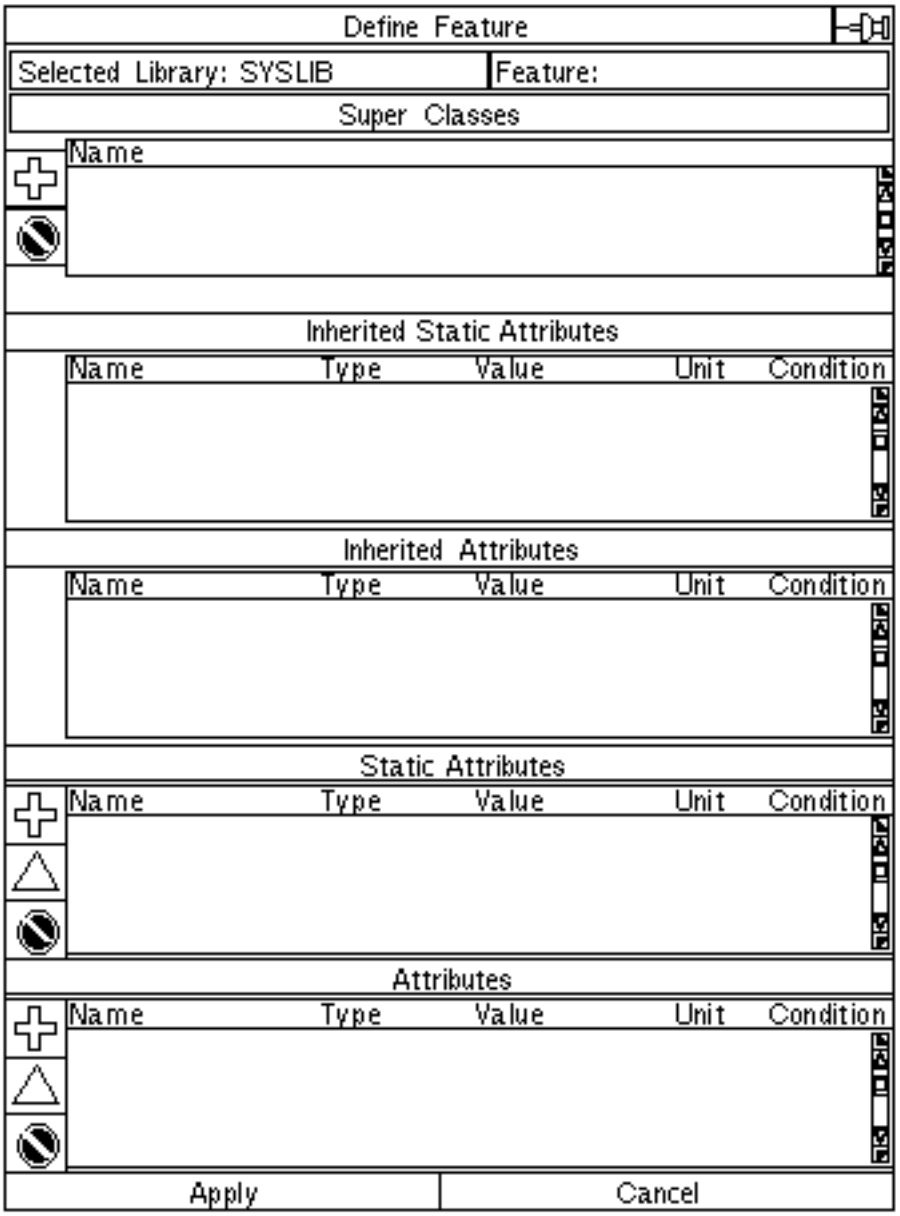
Displays the Select Library property sheet. The library you select from that property sheet plus the system library comprise the virtual library.



Displays the Define Feature property sheet, which is shown on the next page. The fields within that property sheet are as follows:

- The Selected Library field displays the current user library and accesses a runtime list from which you can choose another library.
- The Feature field allows you to enter a name for the feature you are currently defining and provides access to a runtime list of existing feature definitions within the system library and the user library you choose.
- The Super Classes field provides a runtime list of feature definitions from which you can choose a superclass.
- The Inherited Attributes field lists static and dynamic attributes inherited from the superclass(es) of the current definition.
- The Static Attributes field allows you to add, change, or delete a static attribute by means of the plus, delta, and delete icons.
- The Attributes field allows you to add, change, or delete an attribute by means of the plus, delta, and delete icons.

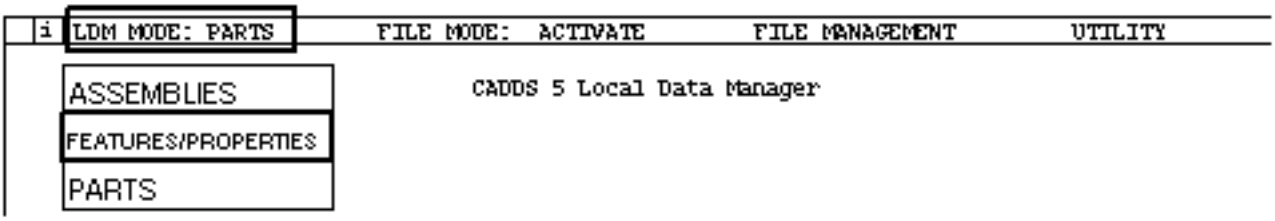
Areas of the Define Feature property sheet are hazed to indicate that you are not allowed to modify a SYSLIB feature definition. If a user library were the selected library, those areas would not be hazed and you would be allowed to modify a definition in that library.



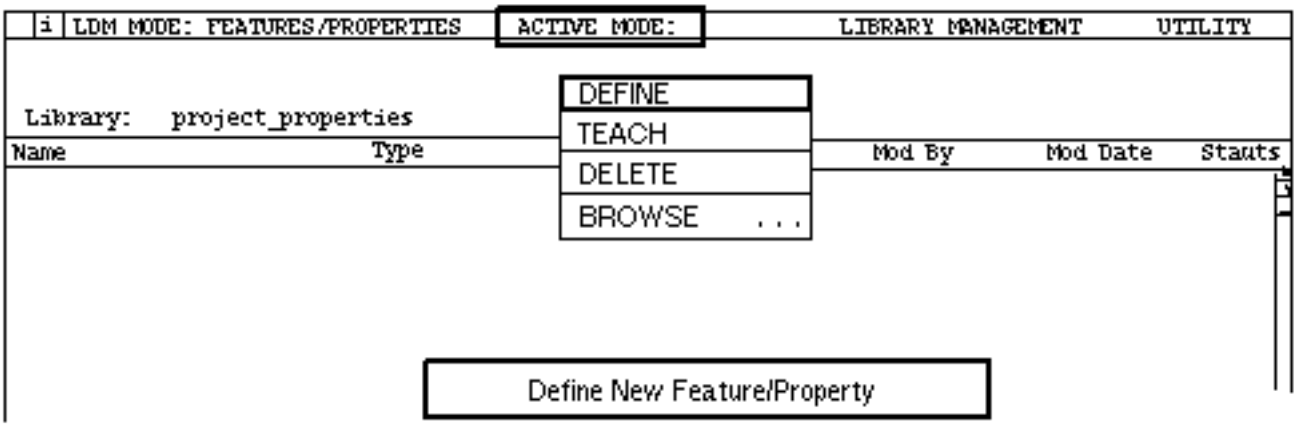
Please note: Once you choose Apply, the system automatically checks to ensure that the current library contains only feature definitions or is empty. If it contains properties, the system does not allow you to apply the property sheet.

You Can Create a Definition in the LDM

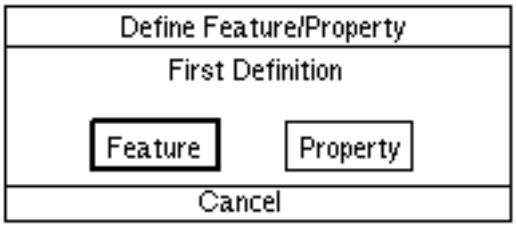
You can also create a definition entirely within the LDM. The LDM includes a Features/Properties page and a Parts page. The system defaults to the Parts page. To display the Features/Properties page, choose the LDM MODE option, and then choose the FEATURES/PROPETIES option from the displayed list.



Once you have the LDM Features/Properties page up, choose Active Mode, choose Define and choose Define New Feature/Property.

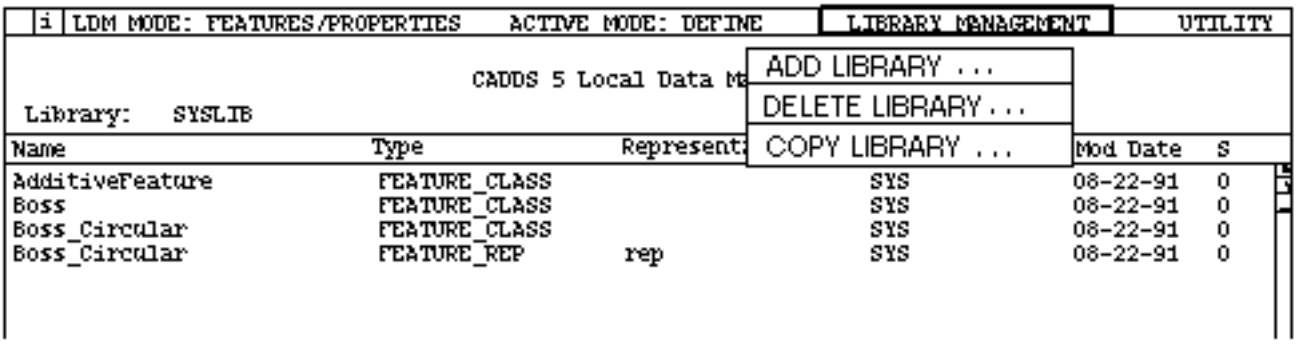


If the current library is empty, the system displays the Define Feature/Property menu. choose Feature and the system displays the Define Feature property sheet.



Library Options in the LDM

By choosing the LIBRARY MANAGEMENT option you can display the options shown in the following illustration:



The Library Management options have the following functions:

- Add Library adds a storage area for features by adding a library to the LDM database.
- Delete Library deletes a library from the LDM database. If you use the Contents modifier, this option deletes a non empty library. Any definitions and representations in the specified library are deleted. Without the modifier, the Delete Library option deletes only empty libraries.
- Copy Library saves or restores a features library to or from LDM or tape. For more details refer to Appendix D, “Setup Procedures.”

Multiple Inheritance

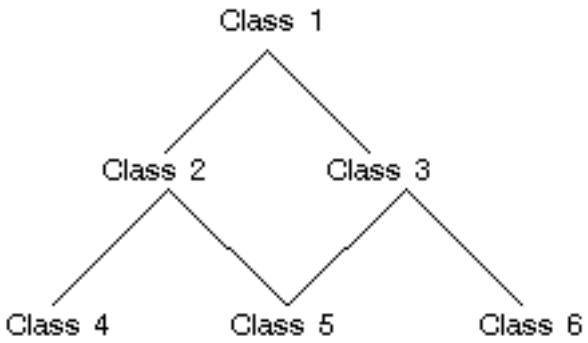
Basic Concept

It is possible for one superclass to have multiple subclasses and for one subclass to have multiple superclasses. The latter concept is called multiple inheritance**.** Multiple inheritance is an advanced object-oriented capability not typically needed but available and useful for complex feature definitions.

Example: In the following illustration,

- Class 2 and Class 3 are subclasses of Class 1.
- Class 4 and Class 5 are subclasses of Class 2.
- Class 5 and Class 6 are subclasses of Class 3.
- Class 5 is a subclass of both Class 2 and Class 3 and thus is a case of multiple inheritance.

Note that the bottom level classes (Classes 4, 5, and 6) are all indirectly subclasses of Class 1.

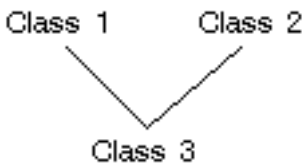


Conflicts across Superclasses

A precedence scheme resolves any conflicts across superclasses. That precedence scheme works as follows:

- If attributes of the same name exist across superclasses, precedence goes to the first specified superclass.

Example: In the following illustration, Class 1 and Class 2 are both superclasses of Class 3. If both Class 1 and Class 2 contain an attribute of the same name, precedence goes to the first superclass specified. Thus if Class 1 was the first to be specified as a superclass, Class 3 inherits the common attribute from Class 1.



- If an attribute of a given name and type exists in a superclass, you cannot specify an attribute having the same name but having a different type.

Example: Assume you are creating a definition called counterbore_hole and specify that counterbore_hole is a subclass of hole. Assume that hole contains an attribute called radius of type integer. You cannot then specify an attribute for counterbore_hole called radius of type length.

- An inherited, dynamic attribute can be overwritten when you are creating a feature definition.

Example: Assume you are creating a definition called tapered_shaft and specify that tapered_shaft is a subclass of shaft. Assume that shaft contains the dynamic attribute length. You can then create an attribute for tapered_shaft called length which takes precedence (that is, overwrites) the default value of the attribute length inherited from shaft.

Utilizing Inheritance

You can utilize inheritance when you define your own features. Try to characterize the feature you wish to create and make it a subclass of existing features that have those characteristics.

Example: If you are defining a protrusion, make it a subclass of existing features that have characteristics appropriate for a protrusion. See Appendix A, "Taxonomy of System Features," which describes system feature taxonomy, to help you choose an appropriate superclass.

The System Superclass Feature

What is the Class Feature?

By default, all system features are a subclass of the class Feature and any feature definition you create is by default a subclass of Feature. You can change that default. Feature contains these attributes: origin, orientation, workpiece, application, operation, and representation. See Appendix A, "Taxonomy of System Features" for a taxonomy chart of system features.

Changing the Default

If you are defining a feature and change the default so that your definition is not a direct subclass of Feature and then select another feature as a superclass, your feature remains a subclass of Feature. This is because the feature you selected as a superclass is itself a subclass of Feature.

If Feature is not an Appropriate Superclass

In some cases Feature may not be an appropriate superclass. In such a case select an appropriate superclass and remove Feature from your definition's list of superclasses.

Example:
If you are defining an additive feature like a boss you should be aware that the default value for the attribute operation in Feature is set to Boolean subtract. A Boolean subtract is an inappropriate application for an additive feature. Thus you should select the feature Boss, which has its attribute operation set to Boolean union, as a superclass. Since Boss is a descendent of Feature, the system would automatically remove Feature from the list of superclasses of the additive feature being defined.

The System Superclass Primitive

When you use the Define Feature option, the system superclass Feature is automatically listed as a superclass of the feature you are defining. If you are using the Define Feature option to define a primitive, simply change that superclass listing from Feature to Primitive. The system class Primitive does not allow the specification of a workpiece and application is always a NoApply.

Creating a Representation

Creating a feature representation allows you to determine exactly what an instance will look like and how that representation will relate to its definition.

Overview of the Process

Conducting a Teach Feature session to create a representation involves the following tasks:

1. Choose the Teach option within the Features page of the Local Data Manager.
2. Choose an existing definition from the list displayed on the Features page.
3. Open a Teach part and then create a geometric representation of the feature.
4. Associate the entities, dimensions, labels, or variable names to the attributes of a particular definition. These associations are stored in the representation.
5. Exit the Teach Feature session.

Associating Attributes to a Representation

When you issue modeling commands during the Teach session, you create entities, dimensions, and perhaps labels or variable names. You must link, by means of a graphical selection, a particular attribute with a particular entity, dimension, or label. For example, you can link the attribute length with a particular dimension.

Be aware of the following when you associate attributes:

- You cannot associate an attribute of a particular type with an inappropriate dimension. For example, you cannot associate an attribute of type length with an angular dimension. If you attempt to do so, an error message is displayed.
- You should complete representation geometry creation before associating workpiece and entry/exit face. If you have edited the representation geometry, list the attributes to check that associations you made with graphical selections are still valid.
- You can use the Disassociate Attribute option to break the association between an attribute and its associated parametric dimension, label, or user dimension.
- When you first enter a Teach Feature session, the default active Cplane is Top and hence the value of orientation is Top. If you then use the Define Orientation option to select another Cplane, that Cplane becomes the value for orientation**.** You can define an orientation any time during a Teach Feature session.

- The representation geometry you associate to the attribute workpiece determines that feature's geometry. Other geometry can exist within the teach part. When you insert an instance only the associated geometry is applied to the workpiece you select. Multiple selections are possible for the attribute workpiece.
- A face attribute accepts only a single face selection.
- You should add constraints to a representation after having associated attributes and associated variables.
- You do not associate non geometric attributes.
- At any time you can determine what is associated to a particular attribute by choosing the Associate Attribute option and selecting an attribute name. The system then highlights either the geometry associated with the face or workpiece attribute or the dimension associated with an attribute of size.

User Dimensions and Constraints

Sometimes the parametric dimensions and labels the system inserts do not directly correspond to an attribute. To address such a case you can do the following:

1. Insert a user dimension that does directly correspond to the attribute and then associate the user dimension to that attribute.
2. Write an equation that involves the attribute and variables you have associated to existing parametric dimensions. You can write the equation such that changing the user dimension triggers the changing of the parametric dimensions.

Once you insert an instance, the constraint equations and variables are hidden, and the user dimension becomes essentially parametric. In fact, an instance displays the user dimension in the configured parametric color. The system remembers the constraint equation set and hence you do not have to maintain it. The "Sample Session: Creating a Feature" on page 3-19 provides an example of this function.

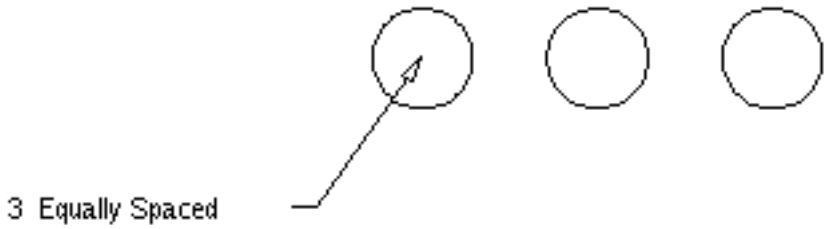
Features Having Parametrically Controlled Groups of Entities

You might want to create a feature consisting of a parametrically controlled variable number of entities. You may want to change the number within that group during feature insertion. In such a case, make sure that during the Teach Feature session, you create the maximum number of such entitles ever likely to be needed.

Example: Suppose you are creating a feature class called Hole_Pattern.

- Among Hole_Pattern's attributes is an attribute called Number.
- The attribute Number is of type integer and is associated to the parametric label that controls the number of holes in the pattern.
- You think that the maximum number of holes in Number's pattern need not be more than 3.

You should construct a pattern of holes having a parametric label, associated with the attribute Number, that has a value of 3 or higher. When you insert Hole_Pattern, you can change the value of attribute Number only up to that maximum of 3 which you constructed during the Teach Feature session. The following illustration shows what the representation for Hole_Pattern might look like.



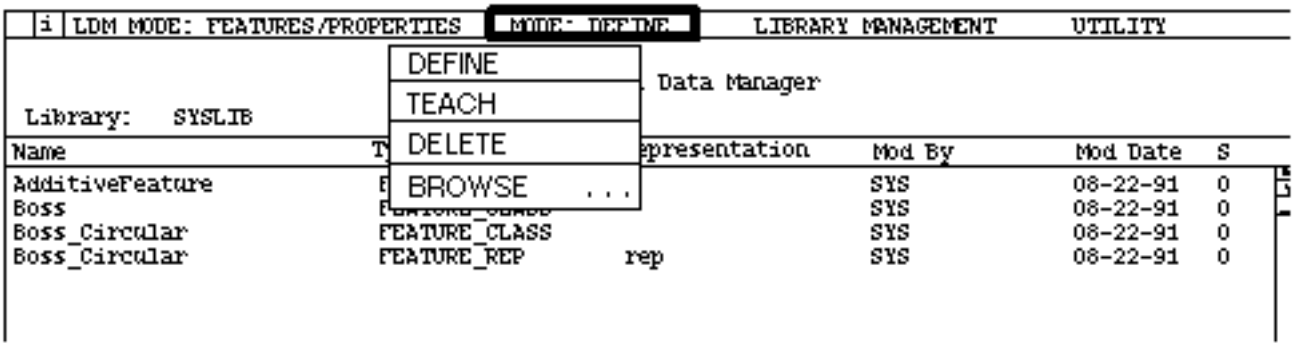
Exiting a Teach Feature Session and Locking

When you exit a Teach Feature session, the Lock/Unlock property sheet appears. That property sheet lists any non associated attributes for the just-exited representation and allows you to lock or unlock the representation. Locking prevents you from inserting an instance of the representation. The system initially locks new representations and representations into which you have copied a part or existing representation.

Options Related to Creating a Representation

Teach Feature Options within the LDM

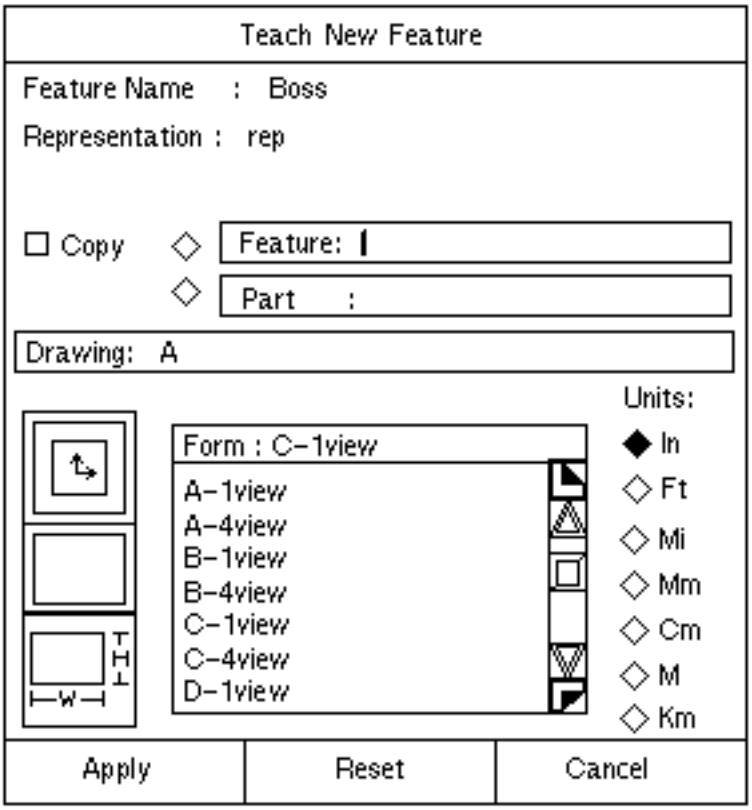
Two of the Feature Mode options, shown in the following illustration, are related to creating a representation. Those options are TEACH and DELETE.



Teach Feature

If you choose the TEACH (Feature) option and then choose a user definition that has no representation, the system displays the Teach New Feature property sheet. If you choose the TEACH option and then choose a definition that does have a representation, the system displays the Teach Feature Old property sheet. You can copy an existing representation into the representation you are creating. Likewise, you can copy a part into the representation you are creating. However, that part must not include explicit geometry, feature instances, or any Profile environment geometry whether it has been applied or not.

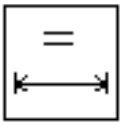
Once you apply the property sheet, a Teach Feature session is activated. The following illustration shows the Teach New Feature property sheet.



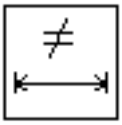
Options within a Teach Feature Session

Displays the Select Library property sheet. The library you choose from that property sheet, plus the System Library comprise the virtual library.

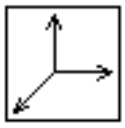
Displays the Define Feature property sheet. Use it to define or modify a feature definition during the Teach Feature session.



Associates an attribute. Choose an attribute name from the displayed runtime list and then choose a dimension, label, or entity. Remember that allowable entities depend upon the named attribute's type.



Disassociates an attribute. Choose an attribute name from the displayed runtime list of currently associated attributes or select all attributes.



Orients the representation relative to the Cplane you select.

Displays the Browse property sheet. The Library field accesses a runtime list of libraries. Once you choose a library, you can use the Feature field to access a list of features within that library. Once you choose a feature, a list of its attributes appears.

Delete Feature

The DELETE (Feature) option allows you to delete a definition or a representation. If you delete a definition, the system automatically deletes its representation (if it has one). You can delete a feature even if you inserted instances of it because DELETE deletes only the global description (its definition and representation) of a feature that exists in the current library. It does not delete the local description created in a part during an Insert Feature action. Instances use that stored local description.

You cannot delete a feature that is a superclass of another property. The system will deny such a deletion request. You can delete a feature that itself has superclasses. The system grants such a deletion request. In those cases the system removes references within a superclass to the class you delete.

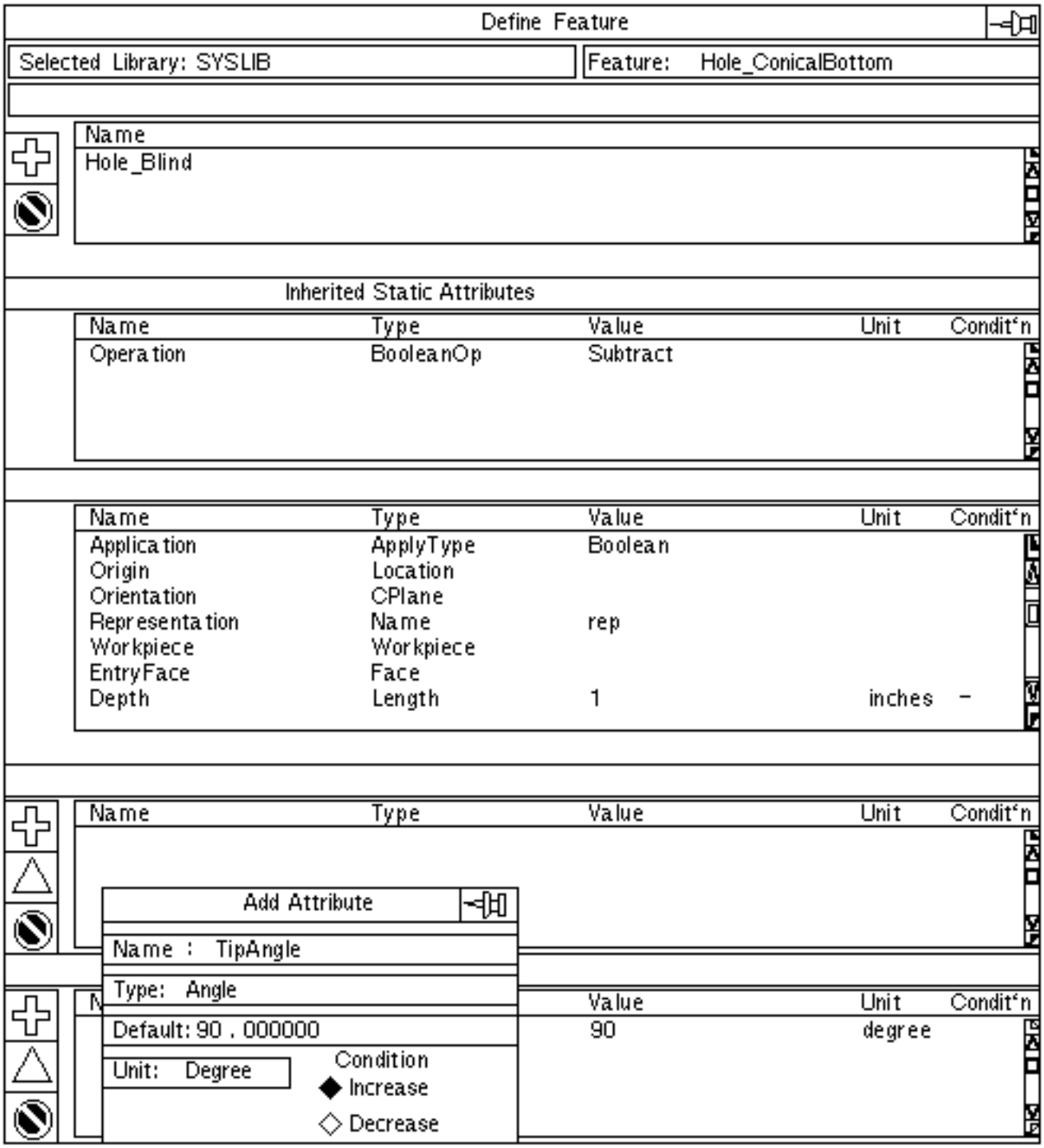
Sample Session: Creating a Feature

This section describes the process by which the system feature Hole_ConicalBottom was created. Both the creation of the definition and the representation are described. Remember that you cannot create a feature in the system library SYSLIB. You must create your own library of features using the Add Library option.

The Definition

Before creating a definition, you should review system feature taxonomy to determine an appropriate superclass. Appendix A, “Taxonomy of System Features” describes the taxonomy for system features. In the case of Hole_ConicalBottom it was determined that an existing feature called Hole_Blind shared its characteristics. Thus Hole_ConicalBottom was made a subclass of Hole_Blind and inherits its attributes. One attribute, TipAngle, was not inherited. It was an addition specific to the definition of Hole_ConicalBottom. The following illustration shows the Define Feature property sheet for Hole_ConicalBottom.

Figure 3-1 The Define Feature Property Sheet



The Representation

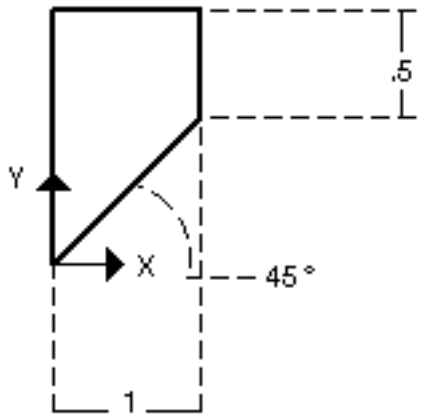
Hole_ConicalBottom, when inserted, is subtracted from a workpiece. Its representation is the geometry to be subtracted. For Hole_ConicalBottom, creating that subtractive geometry involves the following:

- Inserting a rotational sweep
- Adding two user dimensions
- Associating attributes
- Setting constraint equations
- Defining an orientation

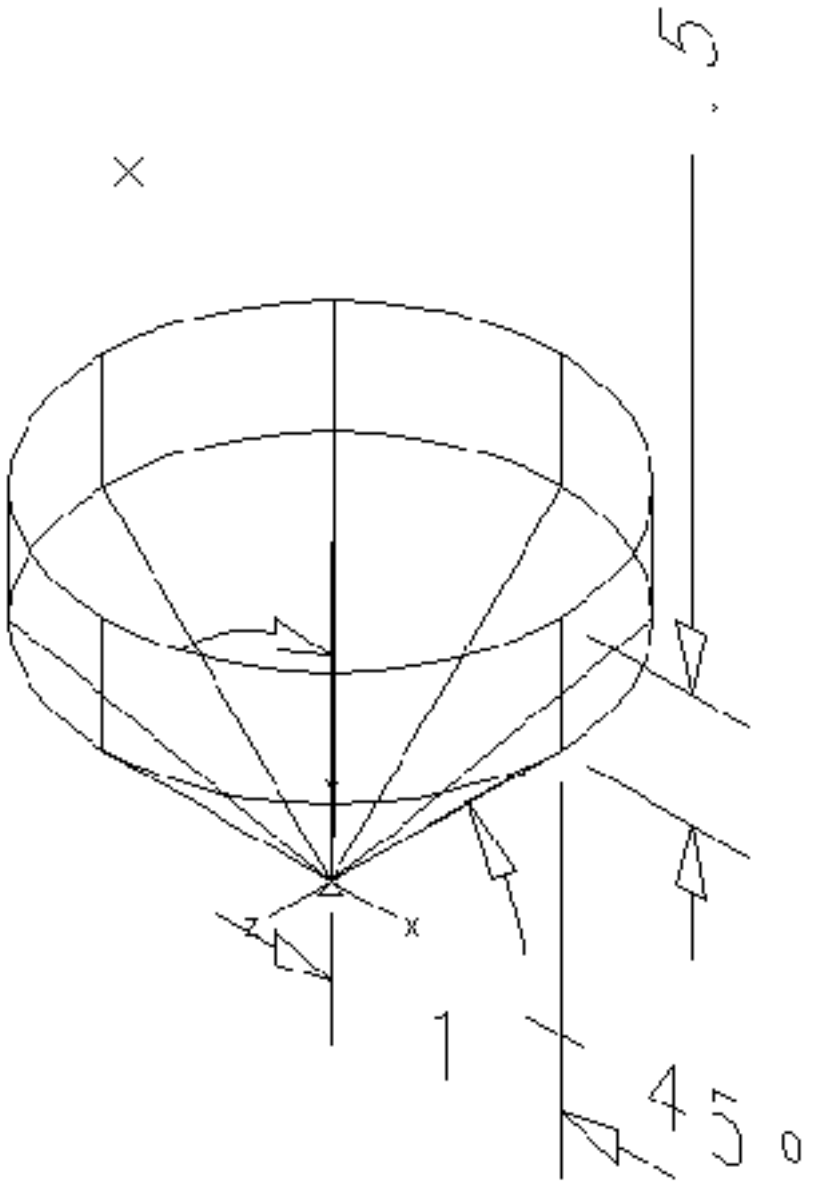
Appendix B, "Command File for Hole_ConicalBottom," lists the commands used during the Teach Feature session for Hole_ConicalBottom.

The Rotational Sweep

A polycurve was inserted and rotationally swept 360° . That profile's dimensions are shown in the following illustration:

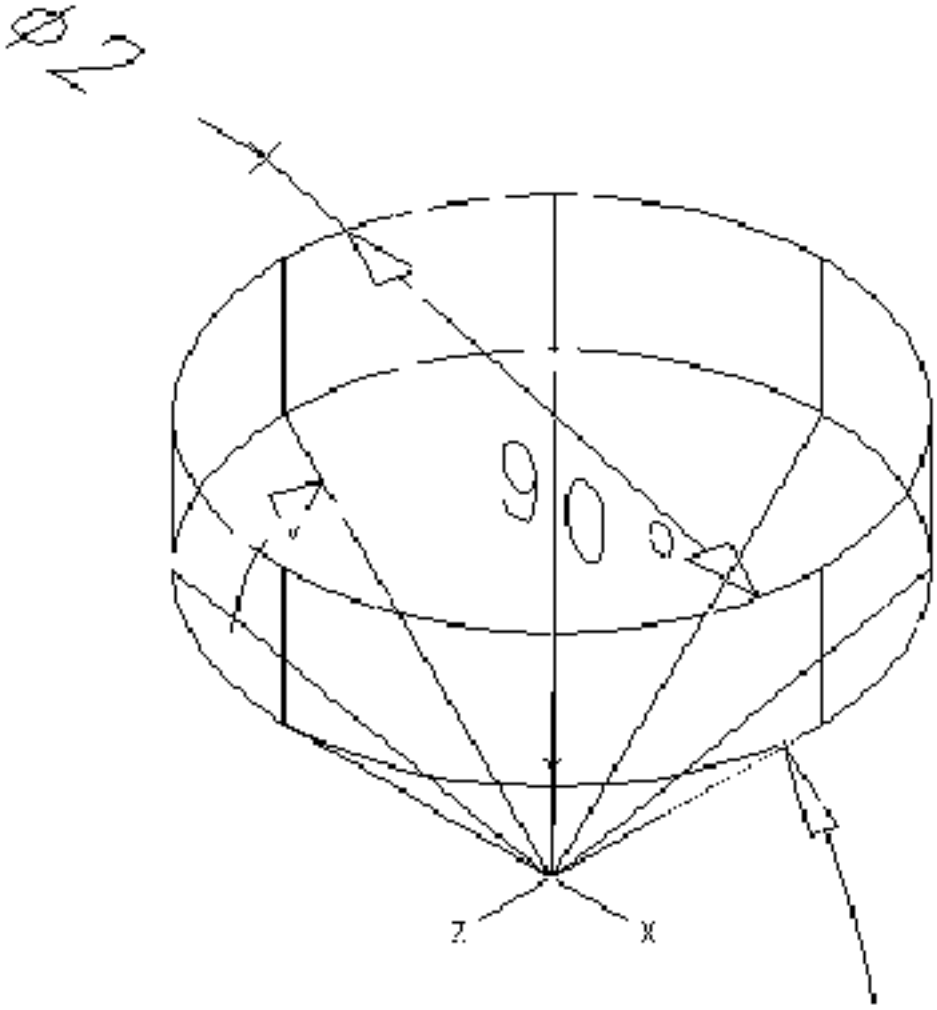


The result of the rotational sweep is shown in the following illustration. Note how the three parametric dimensions correspond to the parameters of the swept polycurve.



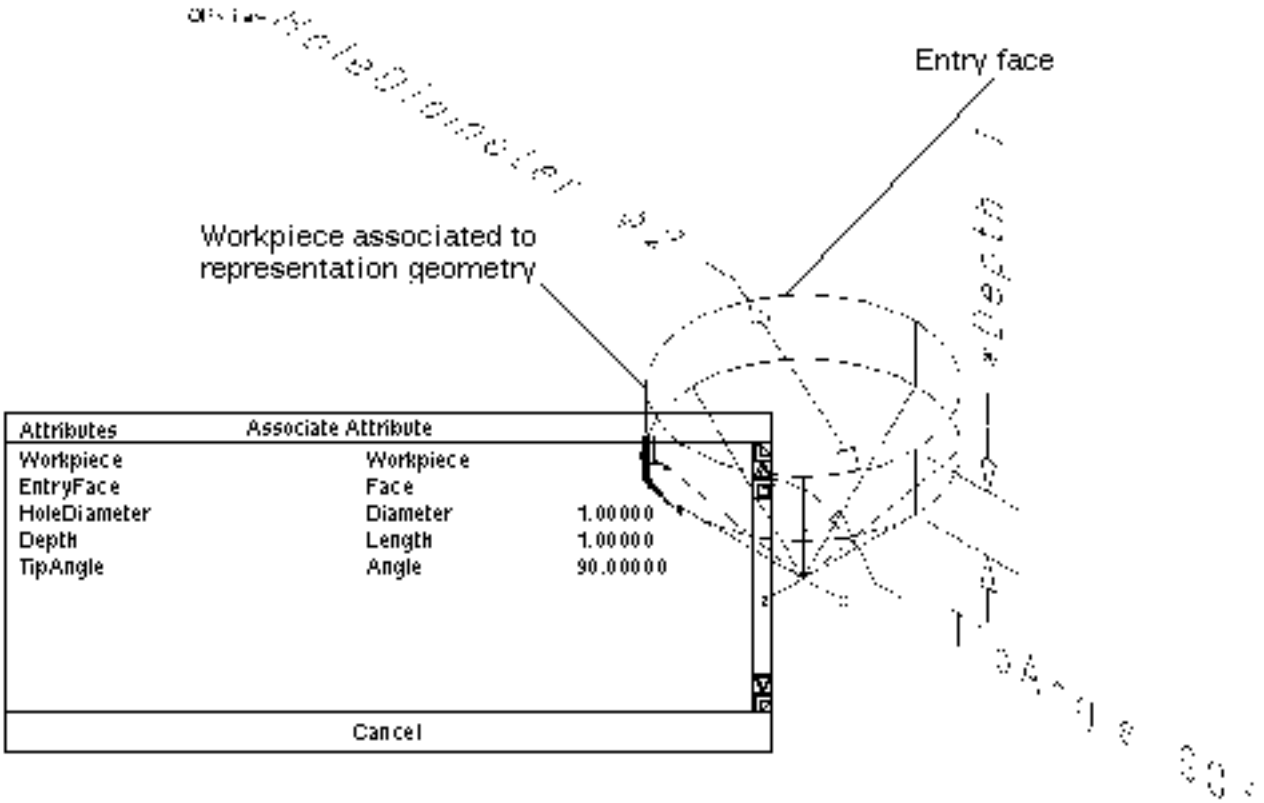
The User Dimensions

The rotational sweep did not result in parametric dimensions that directly correspond to the attributes of Hole_ConicalBottom. No appropriate parameter was inserted for the attribute HoleDiameter (a radial dimension was inserted) nor does one exist for the attribute TipAngle (an angular dimension measuring half the tip angle was inserted). Attempting different modeling options to achieve the needed parameters was not necessary. Instead two user dimensions, appropriate for HoleDiameter and TipAngle, were added. Those user dimensions are shown in the following illustration.



Association of the Attributes

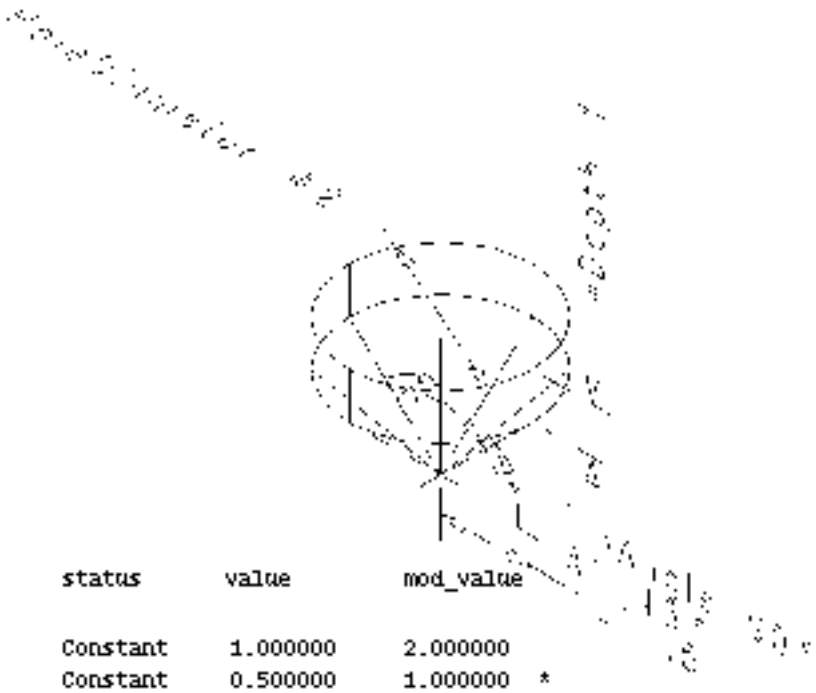
Association of the attributes is the step that links a definition to its representation. The association of the attribute workpiece identifies the geometry to be used when an instance of Hole_ConicalBottom is inserted. Other associations identify the parametric dimensions and user dimensions to be used as driving dimensions of an instance and the face to be mapped to the entry face. The following illustration shows the entity/attribute associations.



The Constraints

In order to make the two user dimensions (the cone diameter and the tip angle) driving dimensions, they had to be involved in constraint equations with the cone radius and the half tip angle. Those constraint equations are HoleDiameter=2*L1 and TipAngle=2*A1. The variable L1 is associated to the cone radius parametric dimension and the variable A1 is associated to the half tip angle parametric dimension.

Once an instance is inserted, the distinction between the parametric dimensions and user dimensions becomes invisible. Since only dimensions associated to attributes are displayed in an instance, A1 and L1 are not visible in an instance. The following illustration shows the representation geometry after the constraint variables are associated and also shows the report window of the variables and constraint equations.



name	type	status	value	mod_value	
HoleDiameter					
	Length	Constant	1.000000	2.000000	
Depth	Length	Constant	0.500000	1.000000	*
TipAngle	Angle	Constant	90.000000	NIL	
L1	Length	Free	1.000000	0.500000	*
A1	Angle	Free	45.000000	NIL	*

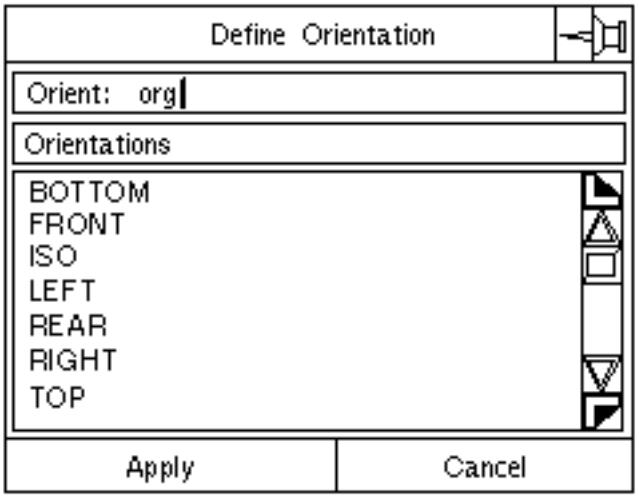
eq_id	text
1	HoleDiameter=2*L1 *
2	TipAngle=2*A1 *

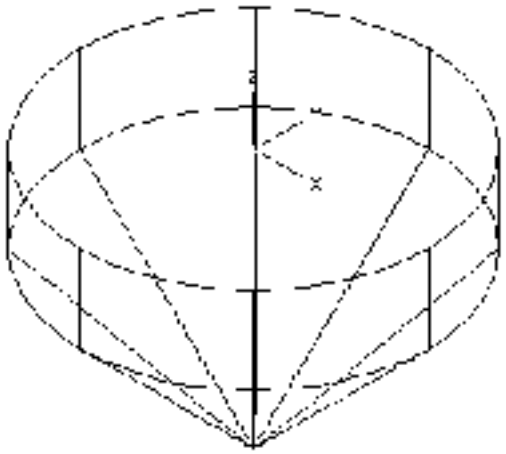
Model Regeneration

Note that at this point in a Teach Feature session, it is a good practice to regenerate the model. Doing so results in a geometric representation that reflects the default values of the attributes.

Orientation and Origin

A characteristic of a hole feature is that its origin is at the center of its opening and that its z-axis runs up and out of its opening. For Hole_ConicalBottom the Define Orientation option was used to place its origin at the location shown in the following illustration.





After a check to assure that all attributes have had associations, the teach part was filed, exited, and unlocked.

Chapter 4

Using Properties

This chapter explains the concept of, creation and use of feature-based properties.

- Concepts
- Creating a Property
- Deleting a Property
- Inserting a Property
- Selecting a Property Instance
- Changing Property Attribute Values
- Making Attachments
- Detaching a Property Instance
- Verifying a Property

Concepts

What Are Properties?

Properties are groupings of attribute information that you can attach to any geometric entity, any feature instance or the part itself. You can easily define such groupings of attribute information. Properties provide a means of adding non-geometric information to the model that you or some application can use.

Properties add descriptive, non-geometric information to a model. In contrast, a feature actually changes the content of a model in that a feature instance either adds (consider a boss) or subtracts (consider a hole) from that model.

Why Use Properties?

Examples of the type of descriptive information properties could add to a model include:

- Material
- Designer
- Mass properties
- Analysis results
- Part identification
- Revision descriptions

Adding such descriptive information results in a more robust design and provides more information for subsequent applications.

How Do You Use Properties?

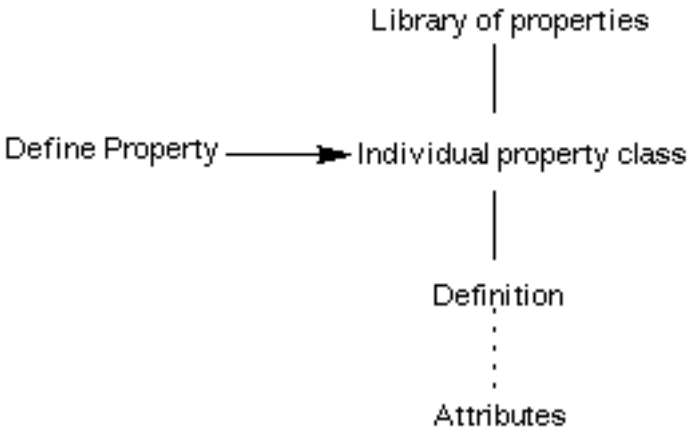
Using properties functions primarily involves:

1. Creating a property class. No system properties are provided, hence all properties must be user-created.
2. Inserting an instance of a property.
3. Attaching the instance to an entity, a feature instance or the part.
4. Retrieving information from an instance.

Creating a Property

The Define Property option allows you to create a property class much like you would to create a feature class.

When you create a new property class with the Define Property option, the resulting property class resides in the current library. The following illustration shows how a property class resides in a property class library much like a feature class resides in a feature library.



Modifying a Class

You can modify an existing property class just as you can modify an existing feature class.

Property Attributes

A property is a collection of attributes. Just as is the case with a feature, a property attribute can be either static or dynamic. Each property attribute consists of the following set of data:

- Name
- Type
- Default value
- Units

Inheritance

The concepts of inheritance and multiple inheritance, discussed earlier in this book, also apply to property classes. That is, a property class can inherit attributes from one or more property classes and a property superclass may have multiple subclasses.

Any property class you create will be a descendant of the root class Property. The root class Property is analogous to the root class Feature discussed earlier in this book.

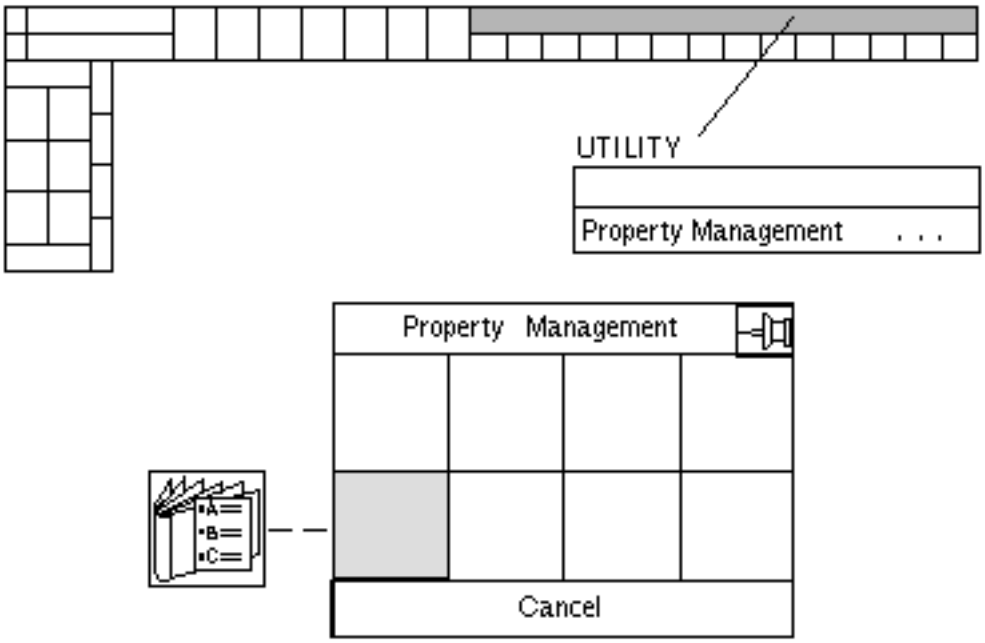
Property Libraries

The system does not allow you to mix properties and features within the same library.

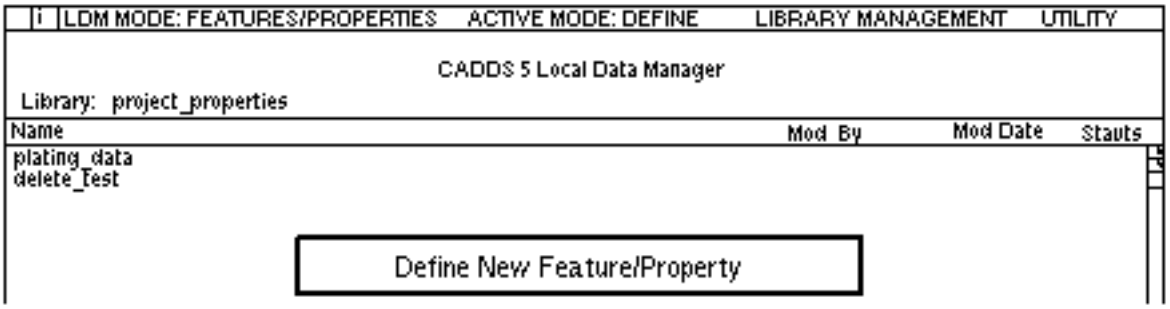
The Define Property Property Sheet

You can access the Define Property property sheet from within the LDM or from within a part as shown in the following illustration. Once you apply the property sheet, the system automatically checks to assure that the current library contains only properties or that it is empty. If it contains feature definitions, the system does not allow you to apply the property sheet.

Within a part



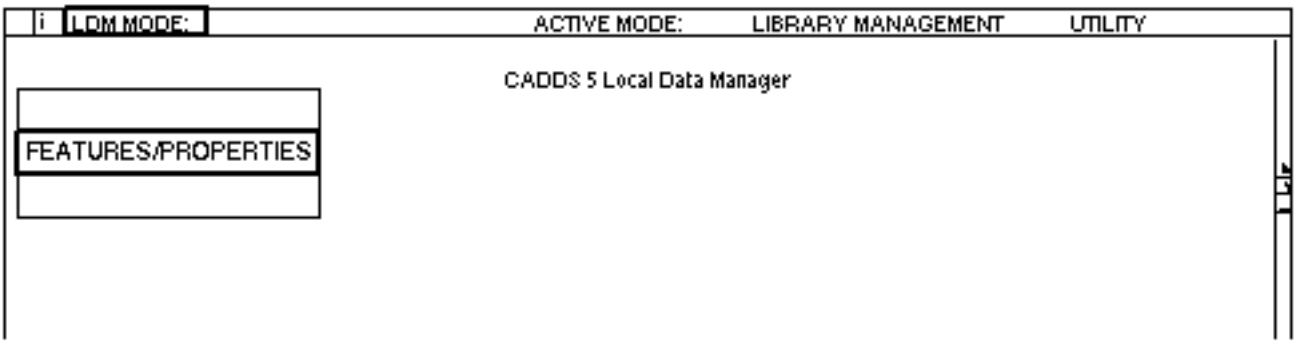
Within the LDM



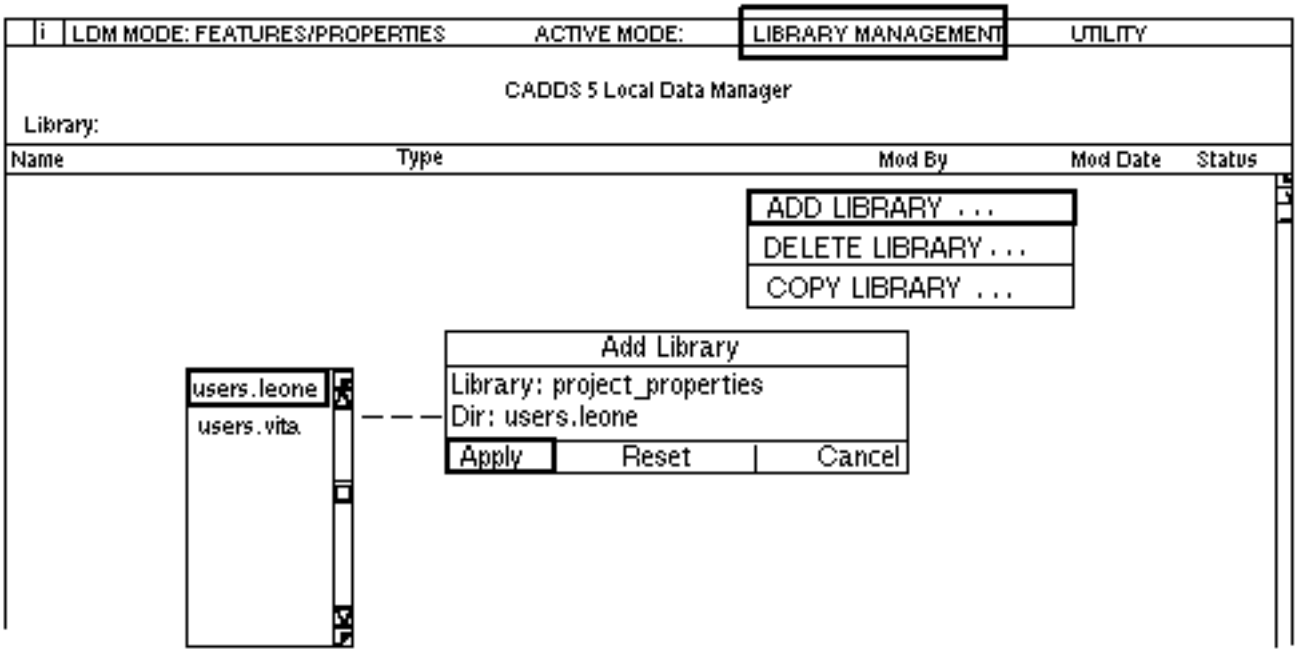
Sample Session

The following procedure creates a property called plating_data. This property, which will includes only a single attribute, is referred to in several subsequent sample sessions within this chapter.

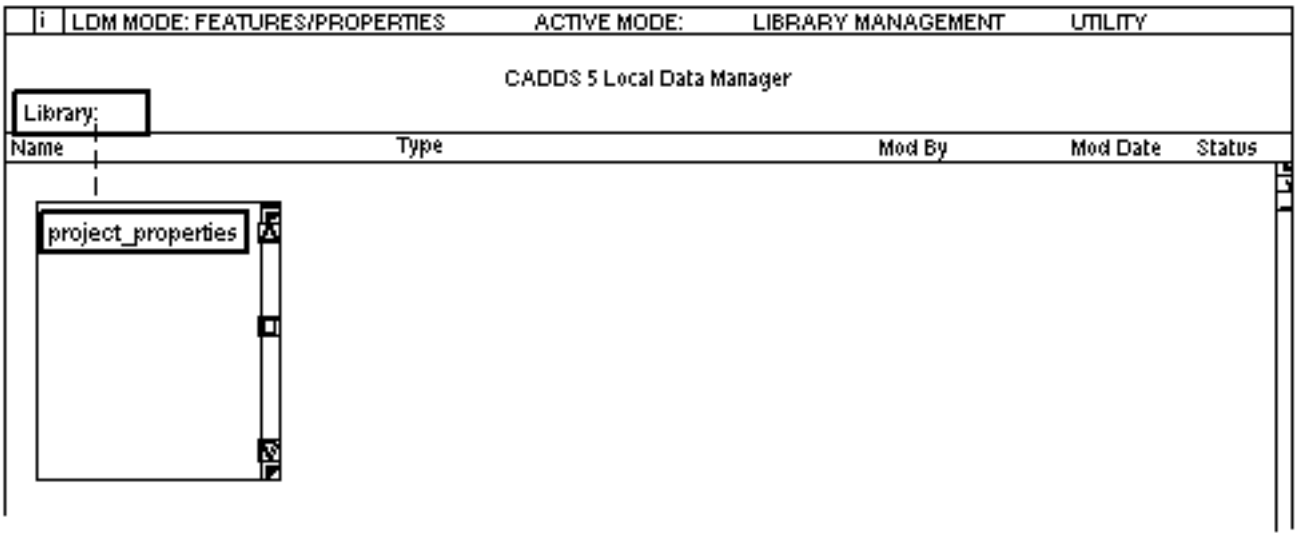
1. Set the LDM to the Features/Properties page by choosing LDM MODE and then choosing the FEATURES/PROPERTIES from the pulldown menu that appears.



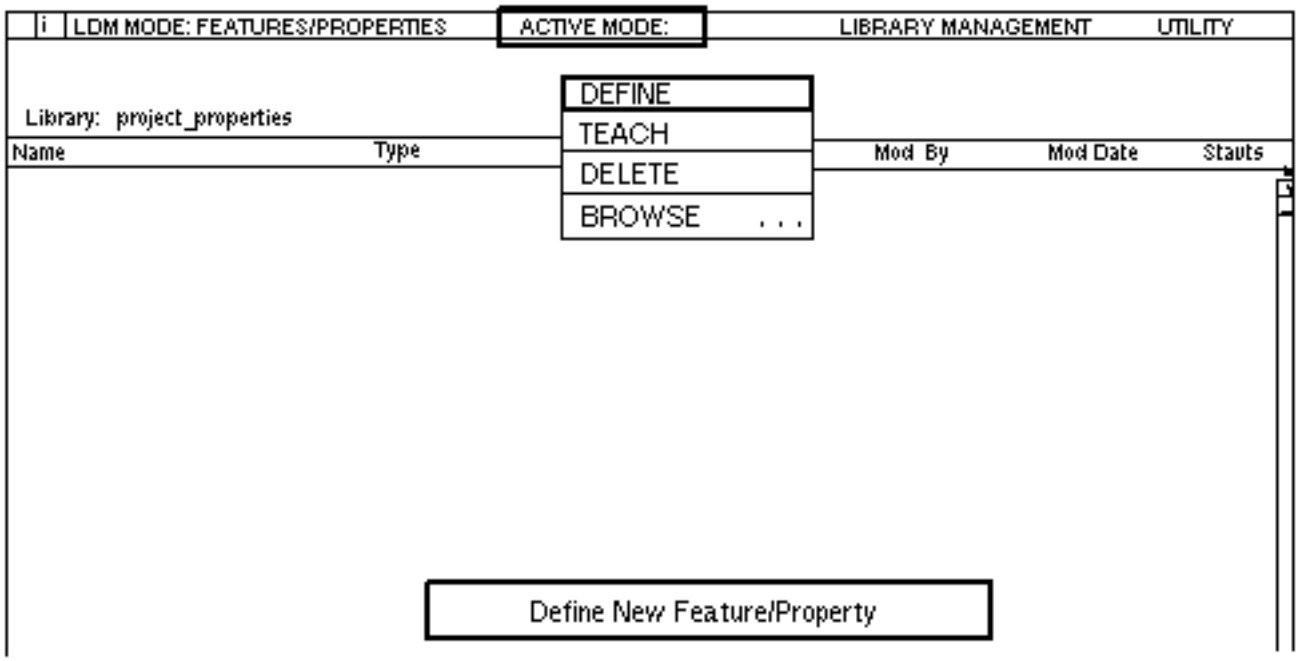
2. You need a library to contain the property you are creating. Choose Library Management, choose Add Library from the pulldown menu and add a library called project_properties. Finally, choose Apply on the Add Library property sheet.



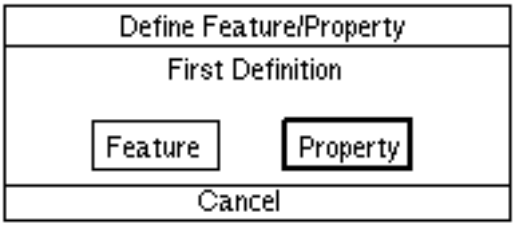
3. Make project_properties the current library. Choose Library and choose project_properties from the scroll list.



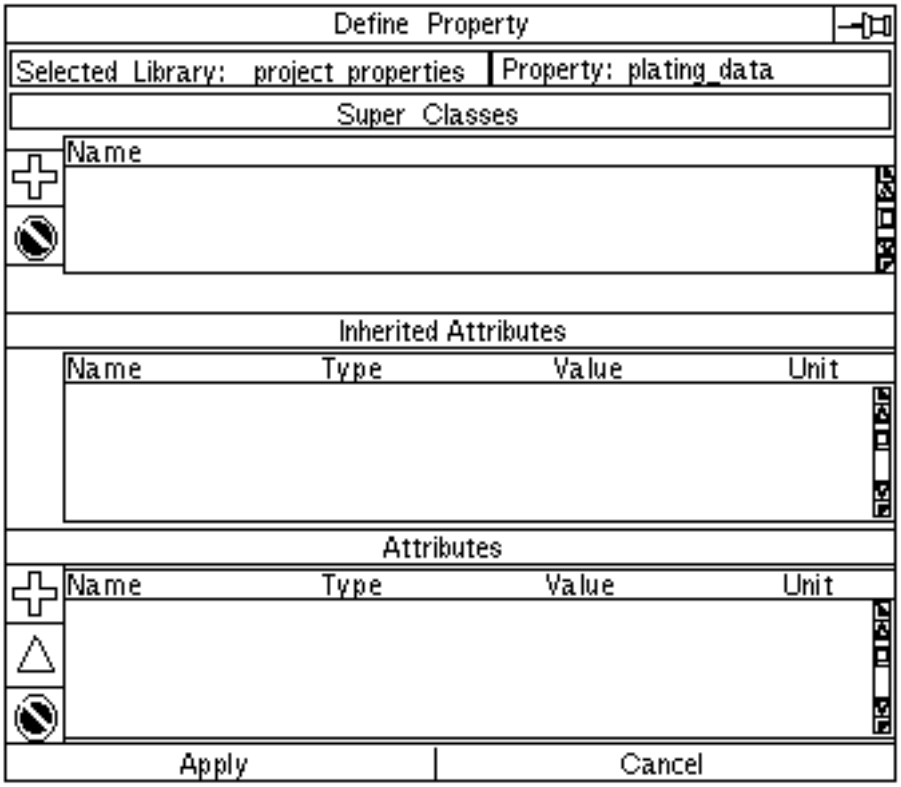
4. You need to define a property. Choose Active Mode, choose Define from the pulldown menu and choose.



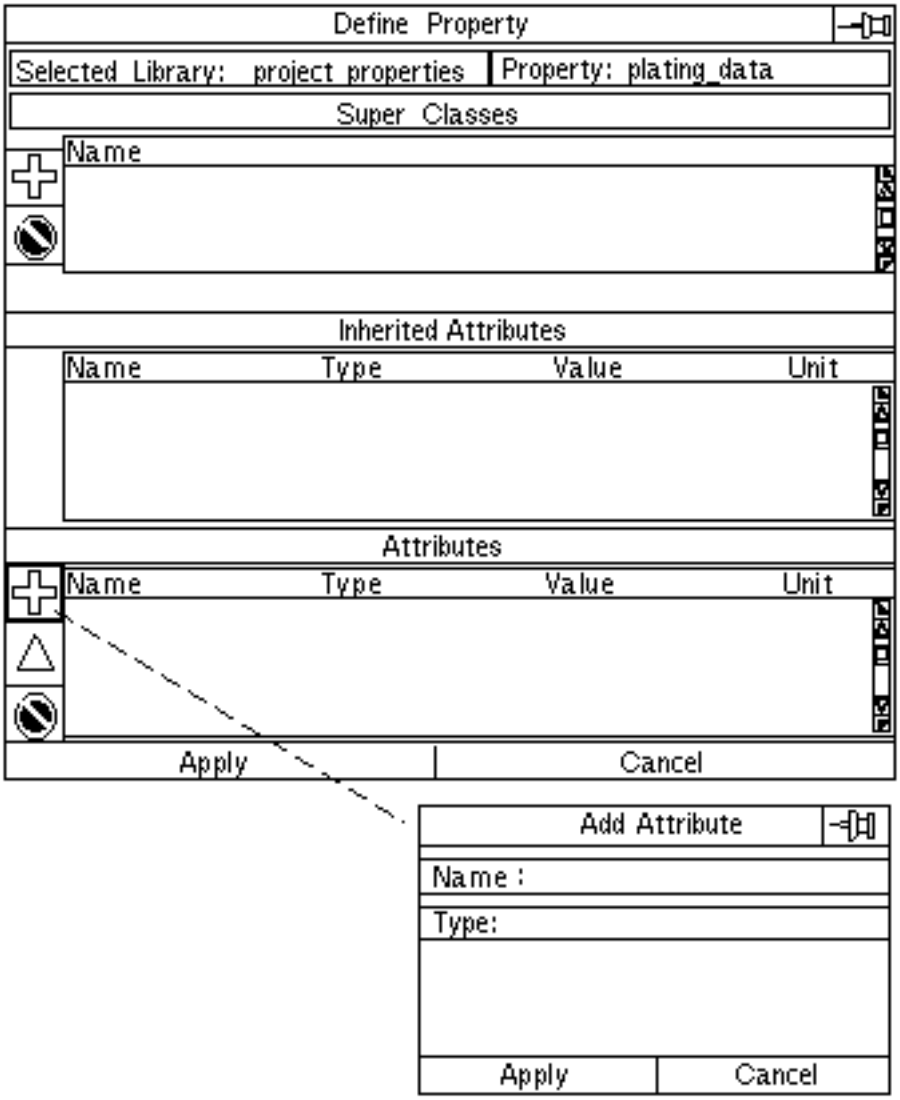
5. Because the current library, project_properties, is empty the system displays the Define Feature/Property property sheet. Choose Property. The system responds by displaying the Define Property property sheet.



6. Specify a property name - plating_data.



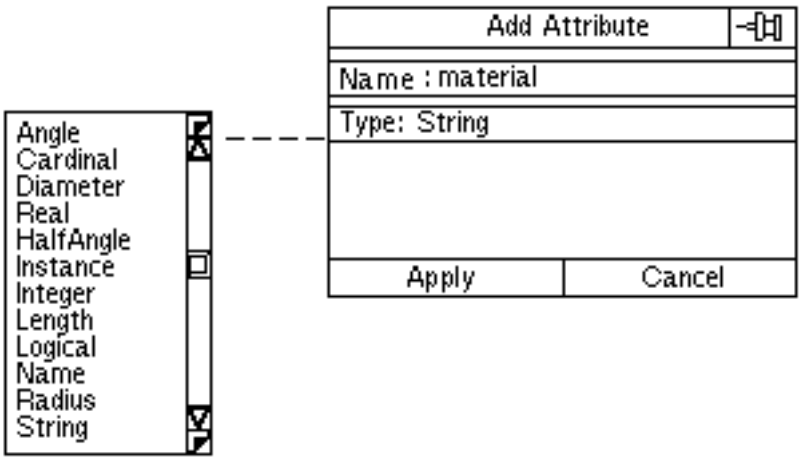
7. Choose the Add Attribute option.



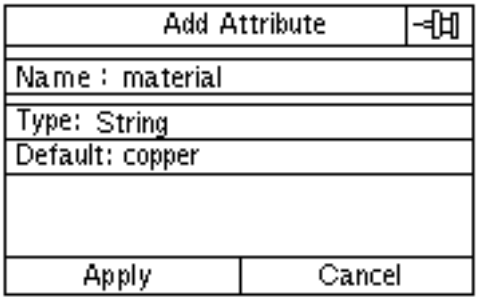
8. Fill in the Add Attribute property sheet as follows:

- Enter the attribute name in the Name field - material.

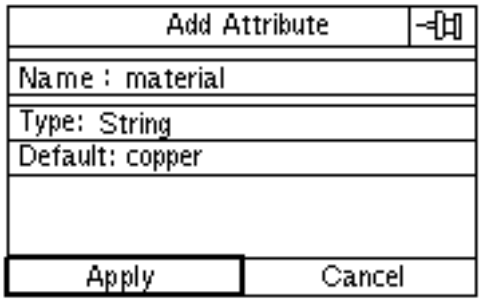
- Choose the attribute type field, Type - and choose String from the scroll list that appears



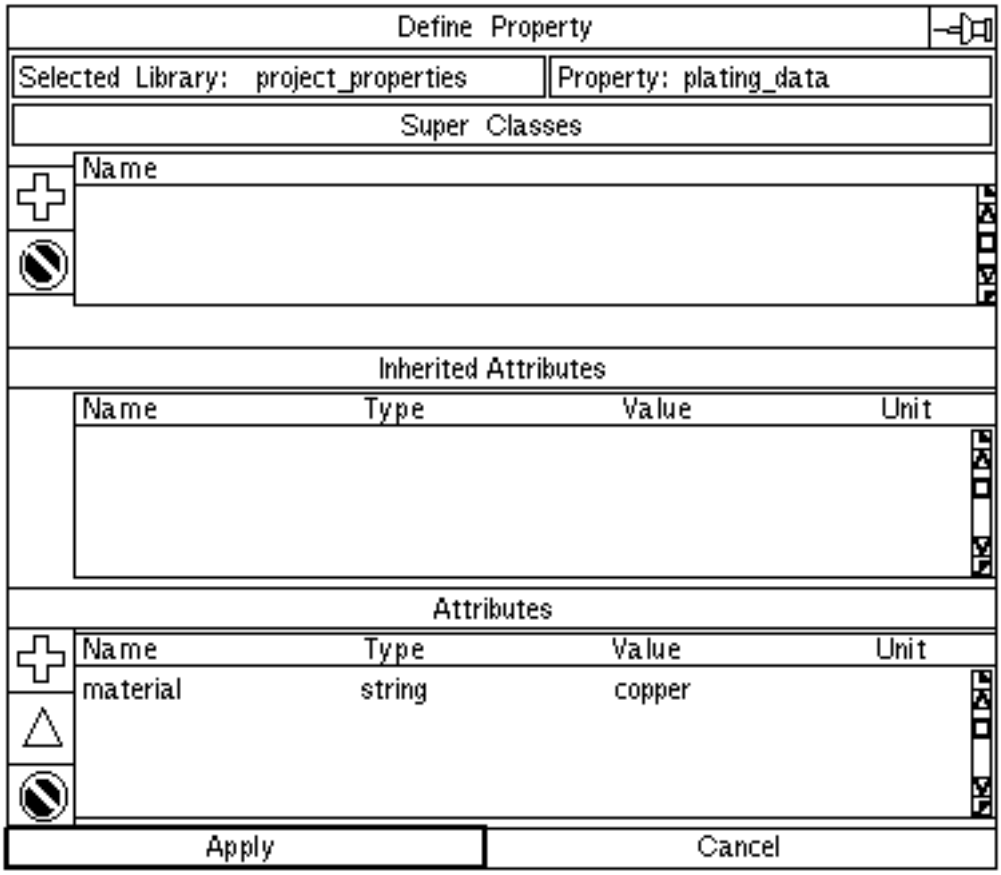
9. Enter the default string for the attribute material in the Default field - copper.

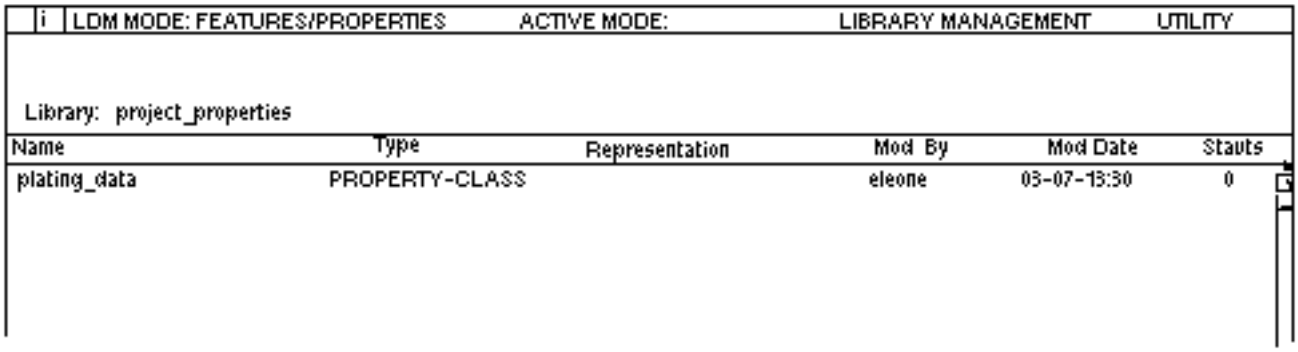


10. Choose Apply on the Add Attribute property sheet.



11. Choose Apply on the Define Property property sheet. The system creates the property plating_data and places it in the current library, project_properties.



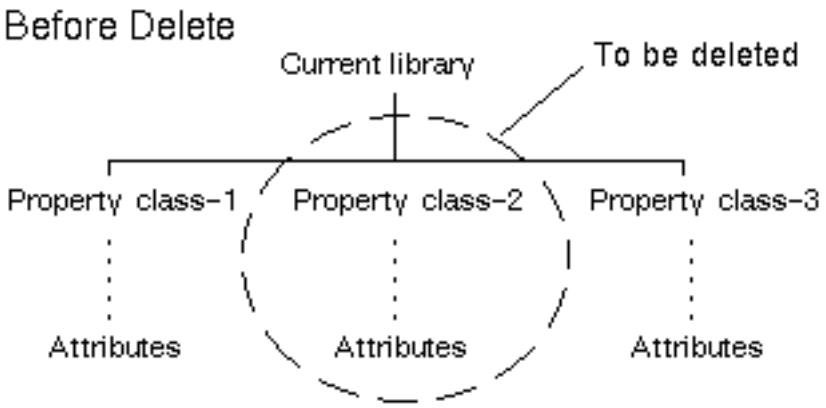


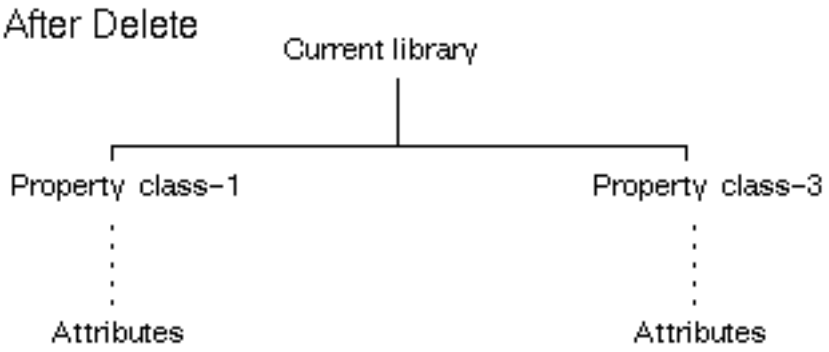
Deleting a Property

The Delete Property option allows you to delete a property class. This option permanently deletes the specified property class definition from the current library.

Example

The following illustrations show before and after a Delete Property action. In this case, Property class_2 was deleted.





What If Instances Exist?

You can delete a property even if you inserted instances of it into the active part. Remember that Delete Property deletes the global description of a property stored in the current library. It does not delete the local description created in a part during an Insert Property action. Instances use the stored local descriptions, not the global one.

What About Superclass?

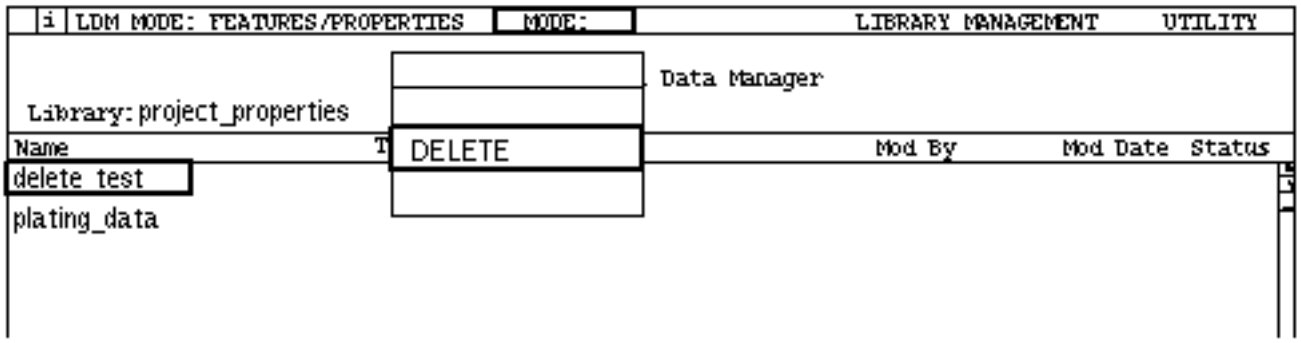
Note the following regarding superclasses:

- You cannot delete a property class that is a superclass of another property. The system will deny such a deletion request.
- You can delete a property class that itself has superclasses. The system grants such a deletion request. In such cases the system removes references within a superclass to the class you delete.

Sample Session

The following procedure deletes a property class called delete_test within the library project_properties.

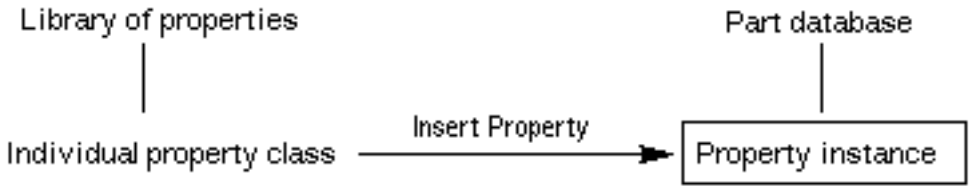
1. Set the LDM to the Features/Properties page by choosing LDM MODE and then choosing the FEATURES/PROPERTIES from the pulldown menu that appears.
2. Choose Mode, choose Delete from the pulldown menu that appears, and then choose the property you want to delete - delete_test.



Inserting a Property

When you insert a property, the system performs two tasks:

1. It inserts an instance of the property class within the current part's database as shown by the following illustration.



2. It creates attachments from that instance to each of your subsequent attachment selections.

Please note: You cannot undo an Insert Property action nor can you insert a property instance during a Teach Feature session.

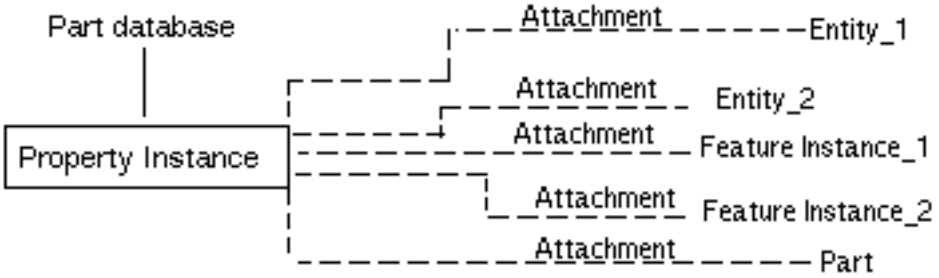
Attaching a Property Instance

The term "attach" means that the system creates a link from the property instance to an entity or feature instance you select or to the part itself. You can make one or more such attachments per execution of the Insert Property option.

You can attach to

- one or more feature instances
- one or more geometric entities
- the part itself

In the following illustration the dashed lines represent attachments (links) between a property instance and a combination of entities, feature instances, and the part.



You Cannot See a Property Instance

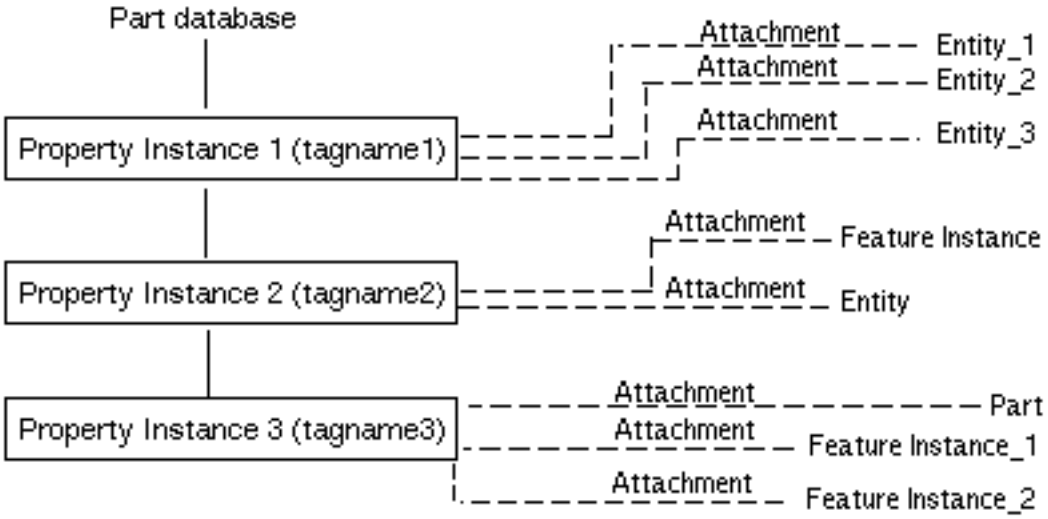
The system does not create a graphic representation of a property instance. That is, you cannot actually see a property instance or any attachments it may have.

Assigning Attribute Values

You can assign new values to the default values of a property's attributes while inserting an instance much like you would for a feature instance. Taking the property created in an earlier sample session, plating_data, you might assign the value "nickel" to its attribute material which was assigned a default of "copper" at definition.

Tagnames

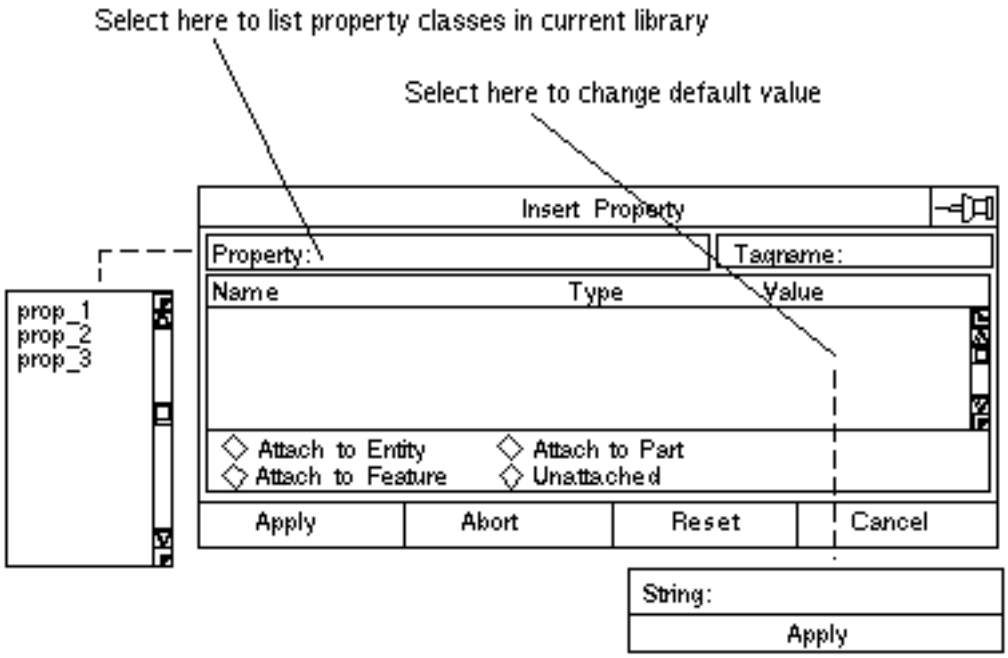
The Insert Property property sheet includes an option for adding a tagname to each instance you insert. You must specify a tagname when using the Insert Property property sheet. Specifying a tagname allows you more flexibility when you need to select property instances. The following illustration shows how a tagname provides a means to uniquely identify a property instance. Note that the tagname references an instance, not an attachment.



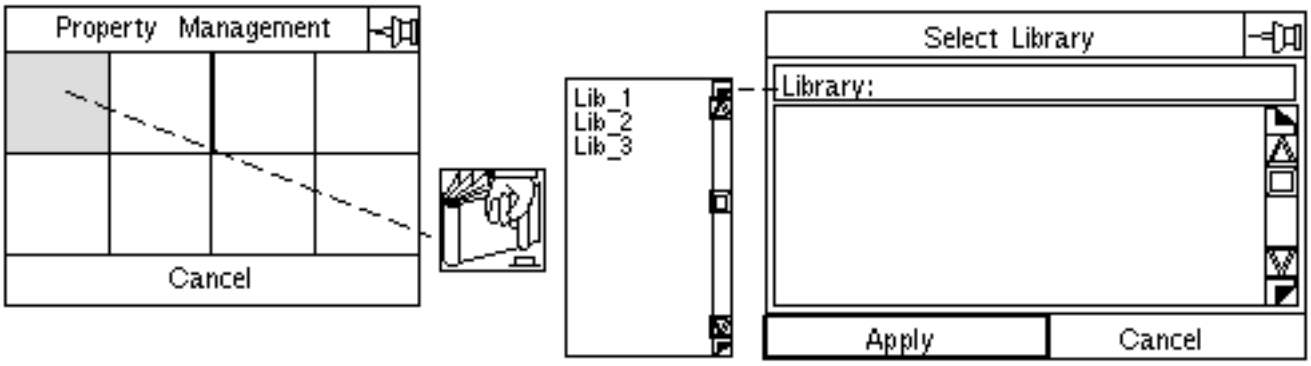
The Insert Property Property Sheet

The Insert Property property sheet allows you to do the following tasks:

- Choose a property class to insert.
- Specify a tagname for the property instance you are inserting.
- Choose an attachment option.
- Change the default value of an attribute(s).



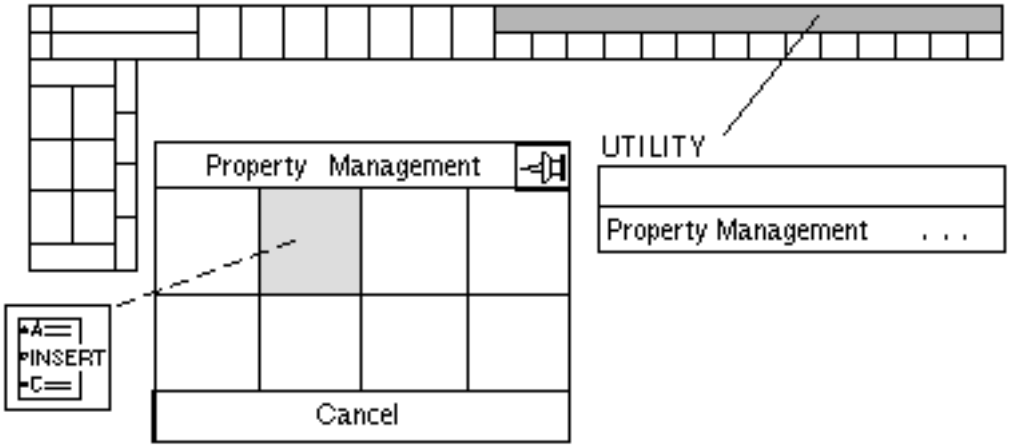
Please note: You can only select a property class from the current library. Use the Select Library property sheet to select another library.



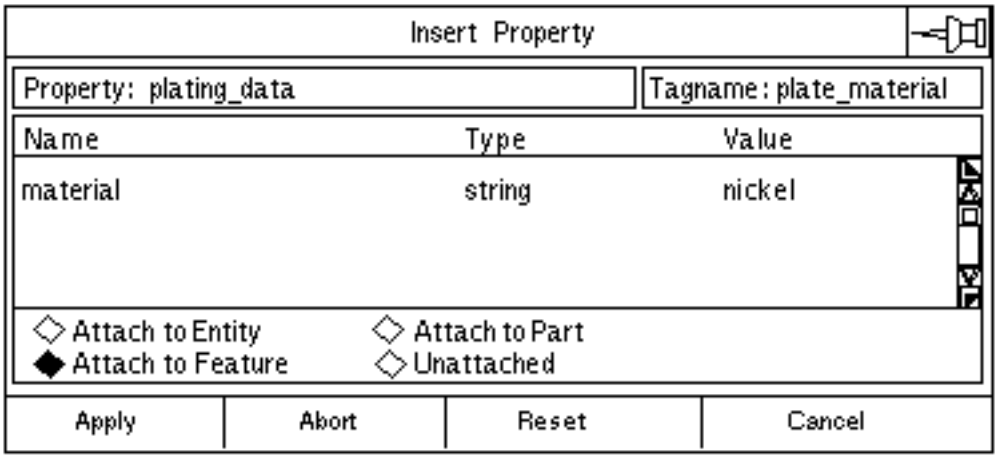
Sample Session

The following procedure inserts an instance of the property created in an earlier sample session, plating_data, and attaches that property instance to an existing feature instance of Hole_Blind.

1. Open a part, then choose Utility from the top menu bar, choose Property Management, and choose the Insert Property option.

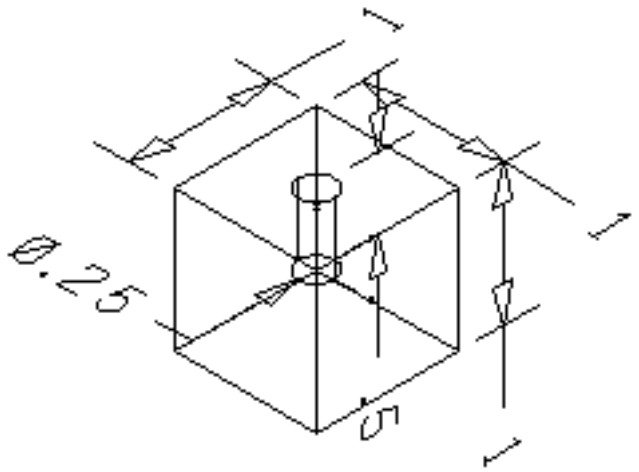


2. Fill out the Insert Property property sheet as follows:
   - Select a property class - plating_data.
   - Enter a tagname - plate_material.
   - Change the default value of the attribute material from copper to nickel.
   - Choose an attachment option - Attach to Feature.

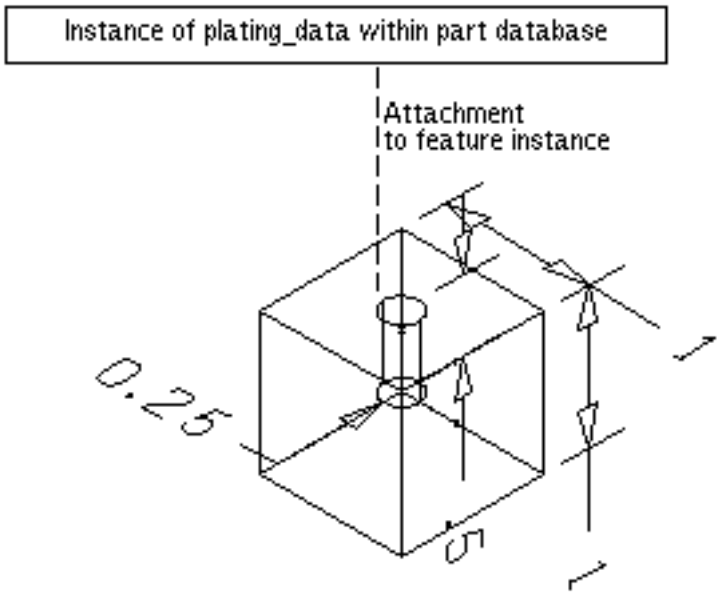


3. Choose Apply. The system now expects you to select a feature instance.

4. Select the feature instance of Hole_Blind shown in the following illustration.



5. Choose Go on the pulldown menu that appears after you choose Apply. The system attaches the plating_data to the feature instance of Hole_Blind and assigns the value “nickel” to the attribute material. The following illustration uses a dashed line to represent the attachment from that instance to the selected feature instance of Hole_Blind.



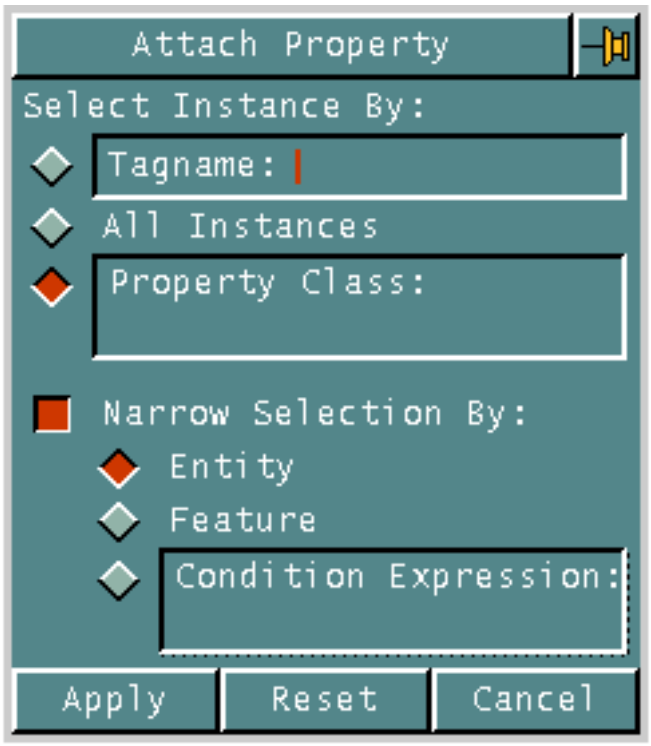
Selecting a Property Instance

Options Requiring Instance Selection

Several options discussed later in this chapter require that you select a property instance. Those options are

- Change Property
- Remove Property
- Attach Property
- Detach Property
- Verify Property Instance

Since you cannot actually see a property instance, their respective property sheets include options by which you can select an instance. The following illustration shows the Attach Property property sheet, which includes the instance selection options Tagname, Entity, Feature, All Instances, Property Class and Conditional Expression. Conditional Expression works only when Property Class is active.



Please note: Instance selection options that appear on property sheets are exclusive of each other. For example, you can choose either Feature or Entity but not both at a time.

Property Instance Selection Options

The following table describes all the property instance selection options. A given property sheet does not necessarily include all of these instance selection options.

Tagname	Selects the instance having the specified tagname.
Class	Selects all instances of the specified class.
All	Selects all property instances in the part.
Entity	Selects all instances attached to the specified entity.
Feature	Selects all instances attached to the specified feature instance.
Part	Selects all instances attached to the active part.
Attached	Selects all instances that are attached to either the part, a feature instance or an entity.
Unattached	Selects all instances that have no attachments.
Condition	Works with Class to select only those property instances that meet the condition set by the specified expression.

More on Selecting by Condition

Conditional expressions are useful to filter out instances you do not want selected. Typically you would create a conditional expression that compares the value of instance attributes with some other value.

Example:
Select only instances of the property class SlotProperties whose attribute TotalVolume is greater than or equal to 100. Such an expression is written: TotalVolume >=100.

Please note: You can use only an attribute having a numerical value in a conditional expression. For example, you can use an attribute of type integer having a value of 21 within a conditional expression.

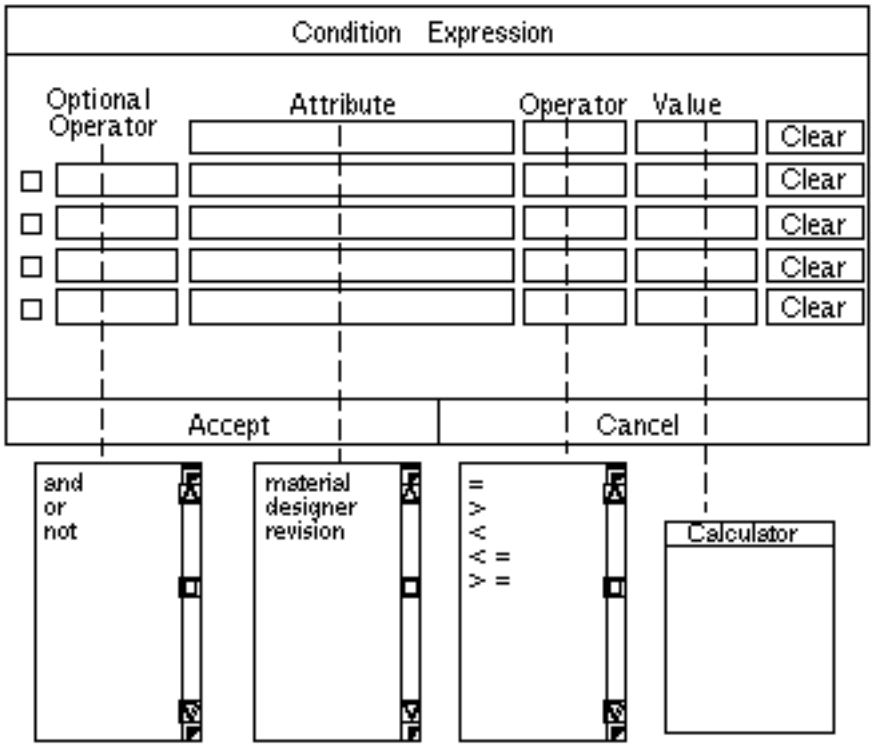
You can use any of the following operators when entering a conditional expression:

- =
- >
- <
- <=
- >=

In addition you can use the following conditional operators:

- And
- Or
- Not

Any property sheet that includes the Condition option allows you to access the following table to specify a condition. Remember, you must have first activated Property Class and chosen a class.



Changing Property Attribute Values

The Change Property option allows you to change the attribute values of an existing property instance. The changes you make are immediate.

This option prompts you to select a property instance. Once you select an instance, the Change Property option then expects you to select an attribute(s) of that instance and enter a new value.

Please note: Like all property options, the Change Property option is not captured by the parametric history and hence cannot be undone.

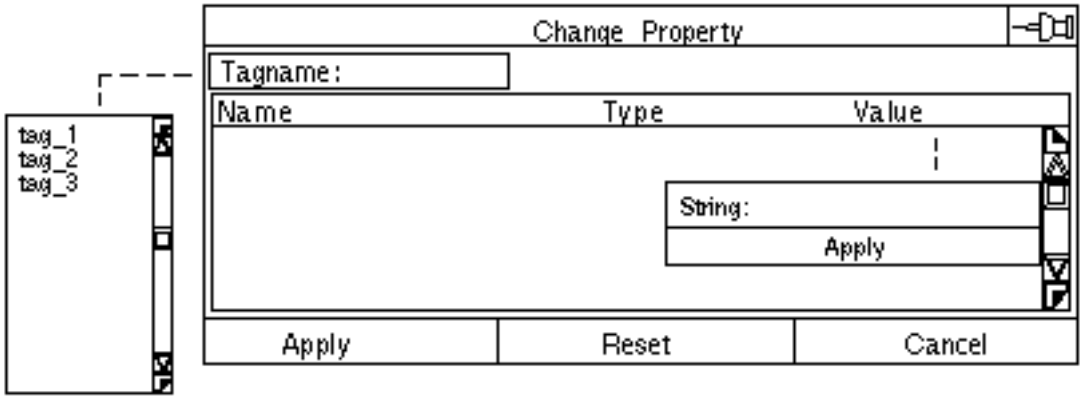
Select Instance by Tagname

You can choose only one instance per execution of the Change Property option. Hence, the Change Property property sheet allows you to choose a property instance by tagname only. That is, the property sheet does not include instance selection options such as Feature or Entity because an entity or a feature instance can have multiple property instance attachments.

The Change Property Property Sheet

The Change Property property sheet allows you to do the following tasks:

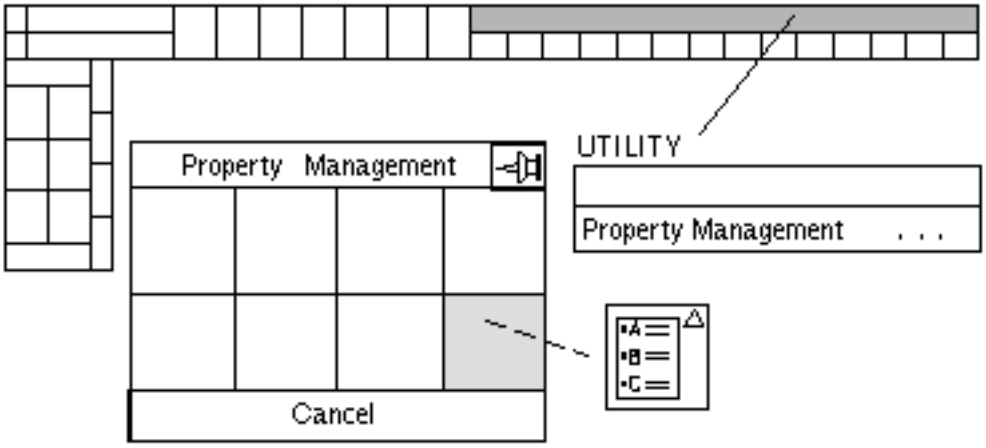
- Choose a property class by tagname
- Change the default value of an attribute(s)



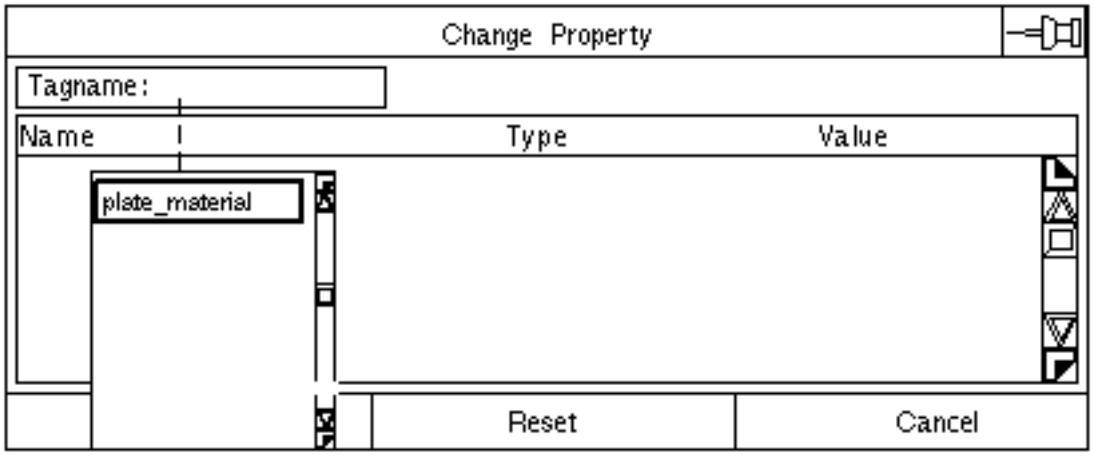
Sample Session

The following procedure uses the Change Property option to change an attribute of an instance of the property plating_data. The instance of property_data was inserted in a previous sample session.

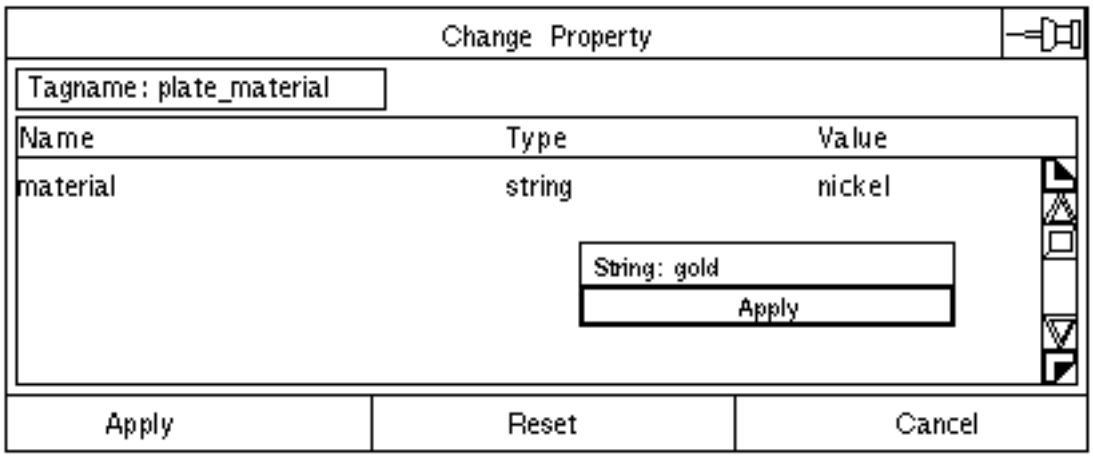
1. Choose Utility, choose Property Management, and choose the Change Property option.



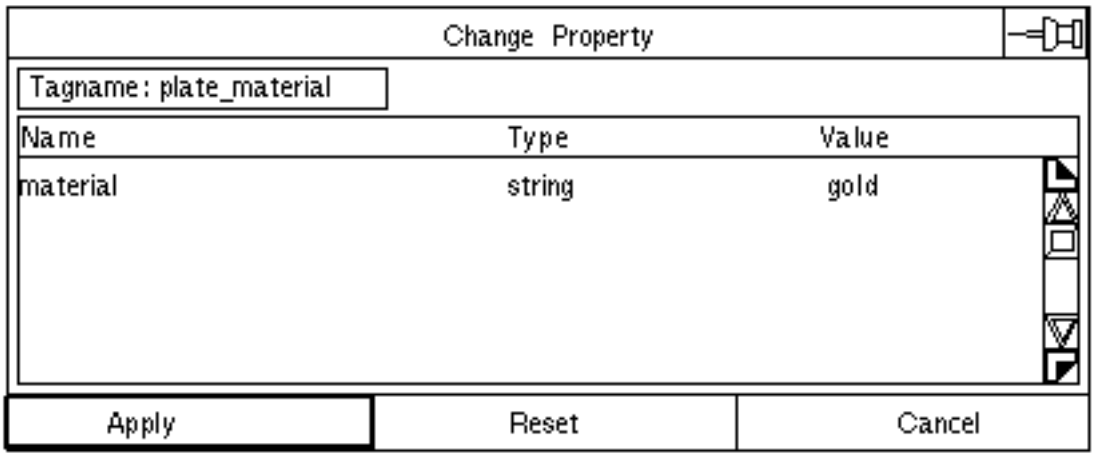
2. Set the Change Property property sheet as shown by choosing the tagname plate_material.



3. Change the value of the attribute material to “gold”.



4. Choose Apply on the Change Property property sheet. This change impacts the instance and its attachments.



Removing a Property Instance

The Remove Property option removes the selected instance from the current part's database.

You can remove only unattached property instances. That is, you cannot remove a property instance that has an existing attachment. The system denies any attempt to do so. If you want to remove an attached instance, you must first detach it using the Detach Property option.

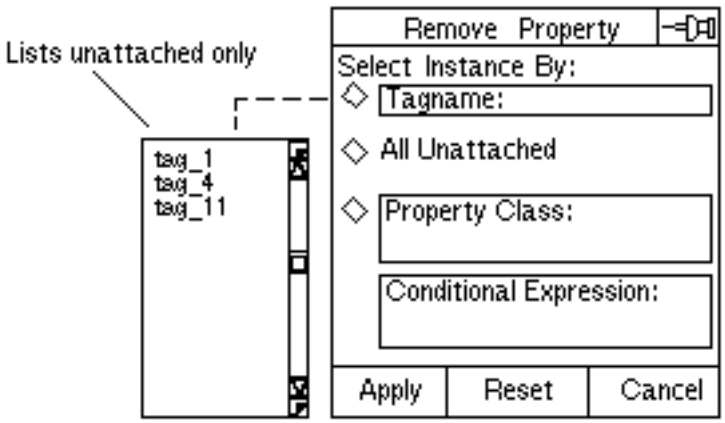
You can remove multiple property class instances per execution of the Remove Property option.

Please note: Like all property options, the Remove Property option is not captured by the parametric history and hence cannot be undone.

The Remove Property Property Sheet

The Remove Property property sheet allows you to

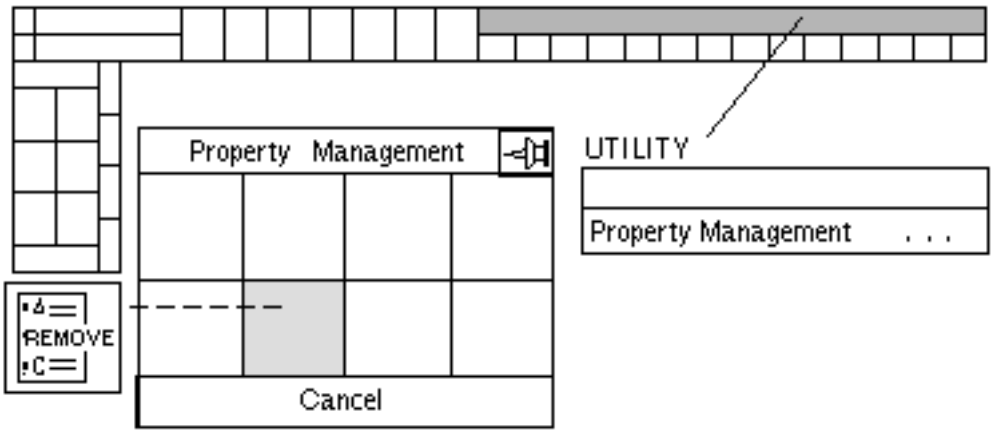
- Select one or more property instance(s) by tagname. The system lists only the tagnames of unattached instances for you to select.
- Select all unattached instances.
- Select all instances.
- Select a property class by class name.
- Write a conditional expression that qualifies the property class name you selected.



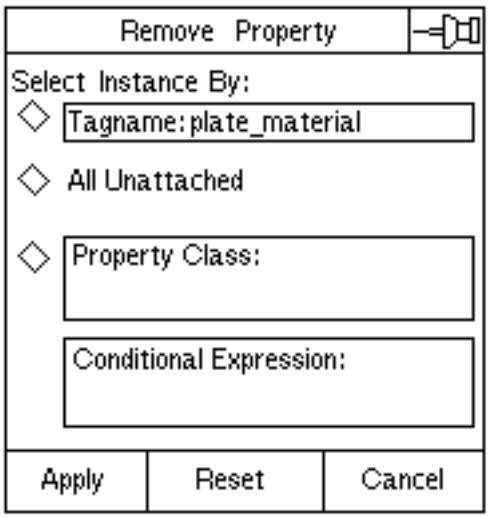
Sample Session

The following procedure uses the Remove Property Instance option to try and remove an instance of the property plating_data. The property plating_data was created, inserted and attached in earlier sample sessions.

1. Choose Utility, choose Property Management and choose the Remove Property option.



2. Set the property sheet as shown by choosing the tagname plate_material.



3. Choose Apply. The system does not allow this removal because the tagname plate_material references an instance having an attachment. (It was established in an earlier sample session.)

Making Attachments

The Attach Property option allows you to create attachments from an existing property instance to a selected entity, feature instance or the active part. You can create an attachment to a combination of entities/feature instances per execution of the Attach Property option, although you must make the instance selections one at a time.

The Attach Property option saves you the effort of specifying attribute values in cases when an existing instance already has the values you want.

In the following illustration, the dashed lines represent the attachments that the Attach Property option creates between an existing property instance an entity and a feature instance.

Please note: Like all property options, the Attach Property option is not captured by the parametric history and hence cannot be undone. You cannot specify attribute values with the Attach Property option.

Multiple Instances and Attachments

You can attach multiple instances of the same property to the same entity. In some cases, such as revision information, multiple instances are useful. You can also attach a single property instance to any combination of entities and feature instances.

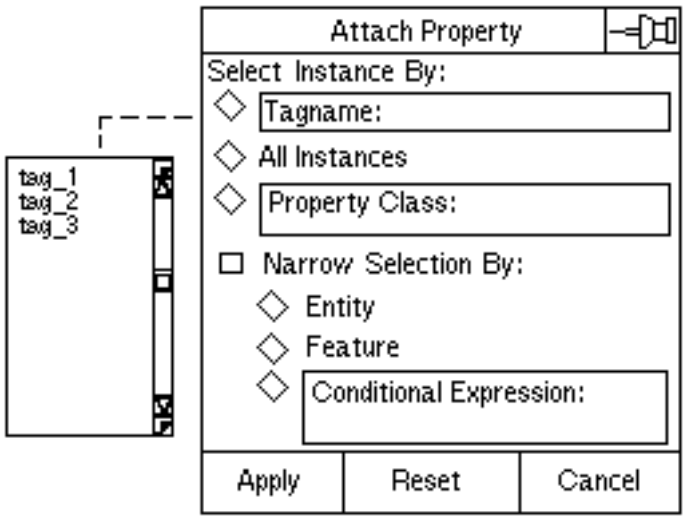
If You Change an Attached Instance

If you change an instance that has attachments using the Change Property option, its attachments recognize those changes.

The Attach Property Property Sheet

The Attach Property property sheet allows you to accomplish the following tasks:

- Choose a property instance by tagname.
- Choose all property instances. You can combine All Instances with Entity or Feature. Doing so narrows your all instances selection down to the property instance that is attached to the entity or feature instance you select.
- Choose a property instance by class name.
- Choose instances that meet certain conditions regarding their attributes. To specify such conditions, you must have first specified a property class. Conditional Expression only works in conjunction with Property Class.
- Choose a property instance that is attached to a feature or entity. Since a property instance can be attached to multiple features or entities, you may want to combine Entity or Feature with Property Class. First choose a property class, then choose the Feature or Entity option. Doing so allows you to choose a property instance of a particular class attached to a particular feature or entity.

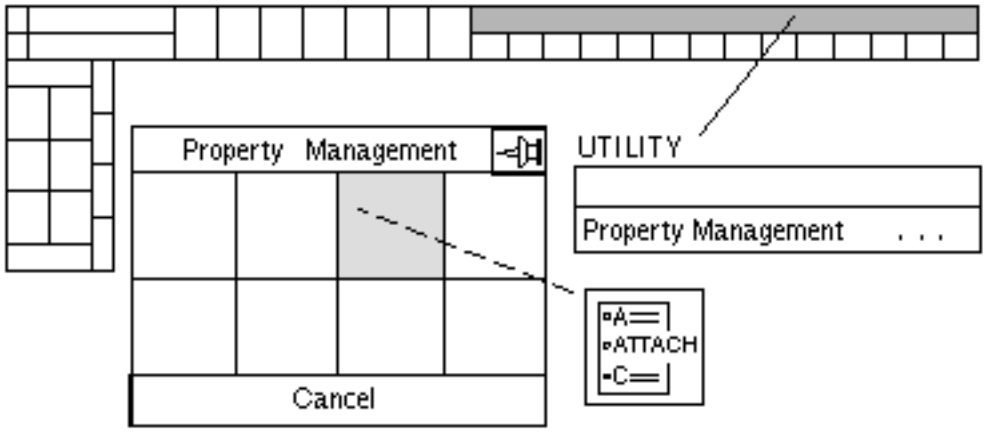


Once you choose Apply, the attachment options, shown in the following illustration, appear. The system then expects you to choose something to which to attach.

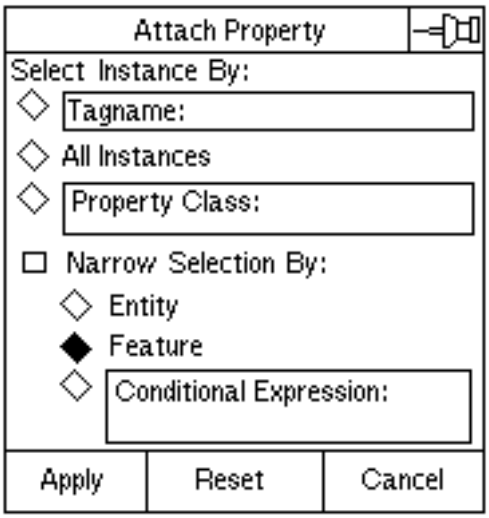
Sample Session

The following procedure uses the Attach Property option to create an attachment from an existing property instance to another feature instance.

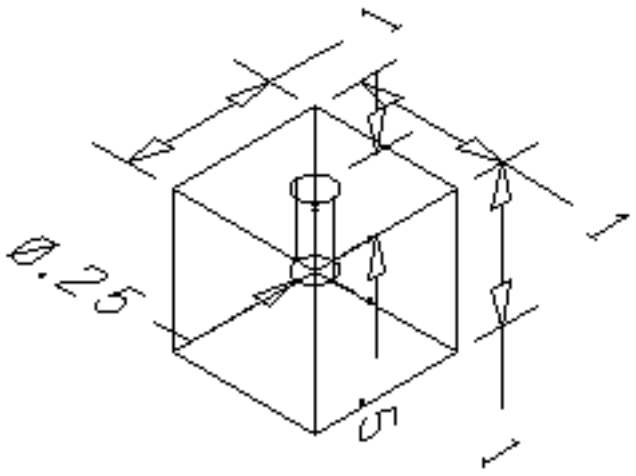
1. Choose Utility, choose Property Management, and choose the Attach Property option.



2. Set the Attach Property property sheet as shown by activating the property instance selection option Feature.



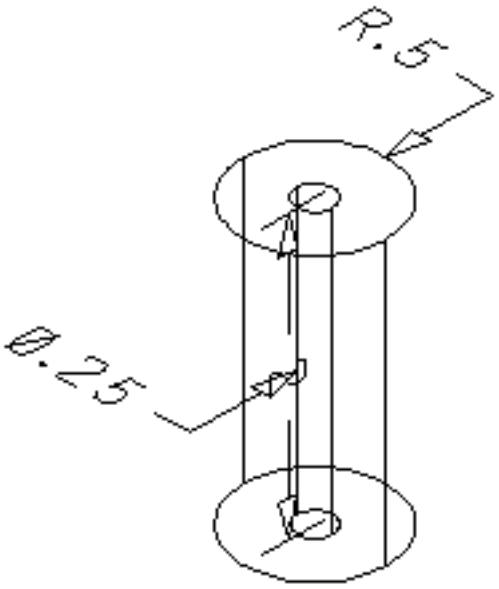
3. Choose Apply. The system now expects you to select a feature instance and also displays several attachment selection options.
4. Select the feature instance of Hole_Blind shown in the following illustration. The system recognizes that an attachment to the instance of the property plating_data exists and thus selects that instance of plating_data.



Select instance of Hole_Blind

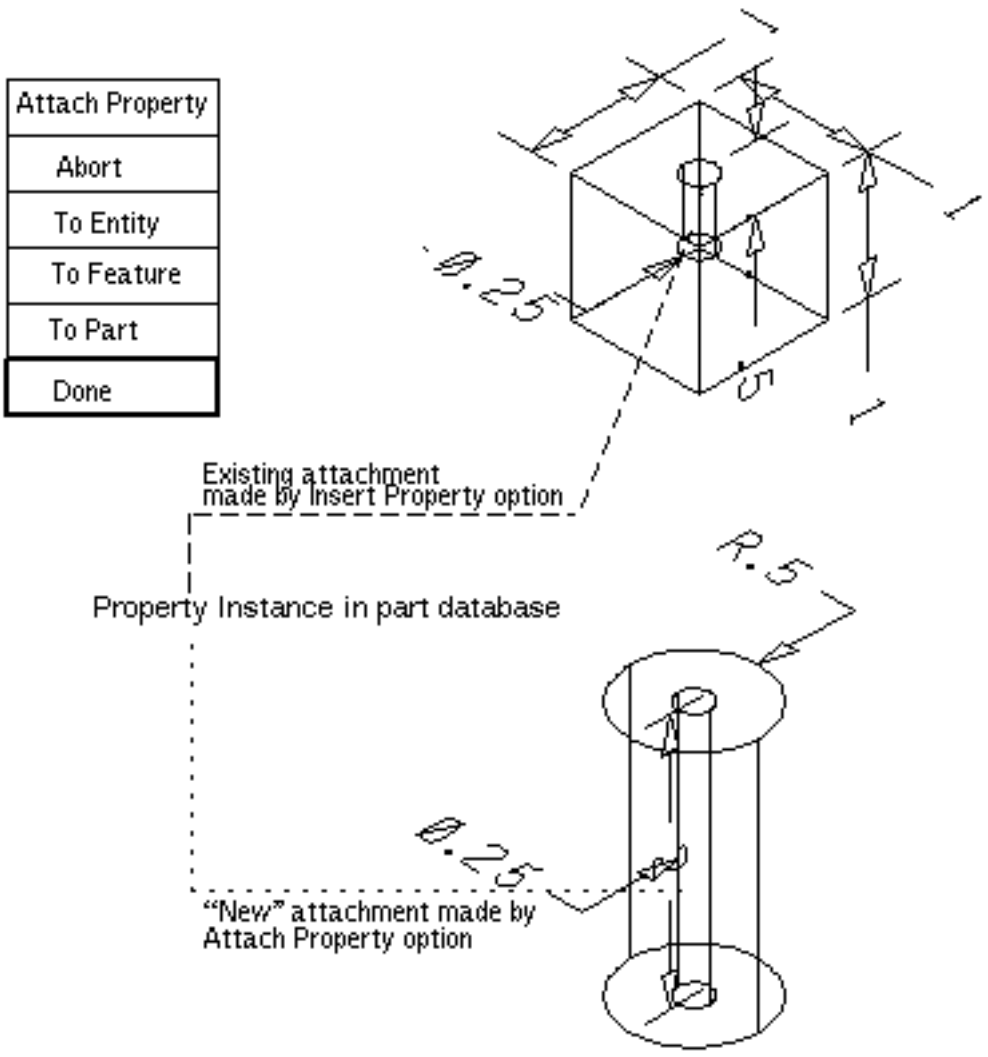
5. Choose the attachment option To Feature. The system now expects you to select another feature instance to attach.

6. Select the feature instance of Hole_Blind shown in the following illustration.



Select this instance of Hole_Blind

7. Choose Done. The system creates an attachment between the instance of plating_data in the part's database and the feature instance of Hole_Blind within the cylinder solid. The following illustration shows the two attachments that now exist: the existing attachment and the "new" attachment you just made.



If You Delete An Attached Entity

The Delete Entity option deletes the selected entity and any attachment it may have. The following illustration shows a property instance with two attachments. Those attachments are represented by the dashed lines.

The following illustration shows the result of using the Delete Entity option to delete Entity_1.

Detaching a Property Instance

The Detach Property option allows you to break the link between an existing property instance and one or more entities within the part. That is, you may detach references you created using the Attach Property option or the Insert Property option. In the following illustration, the dashed lines represent attachments between a property instance an entity and a feature instance. Those attachments may be deleted by the Detach Property option.

The following two illustrations show the result of using the Detach Property option to delete the attachment between the property instance and Entity_1.

The entity and the feature instance themselves remain as does the property instance.

Please note: Like all property options, the Attach Property option is not captured by the parametric history and hence cannot be undone.

If You Detach All Attachments

If you detach all attachments, such that a property instance is not attached to any entity at all, then the system warns you that the instance still exists in the part but in an unattached state.

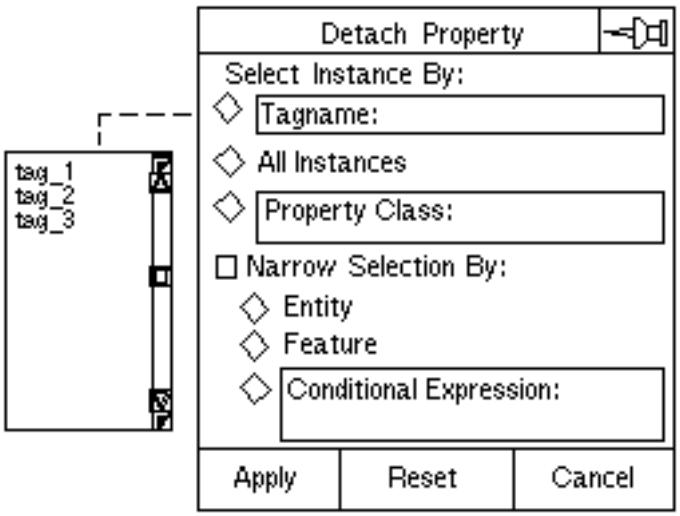
Multiple Detachments

You can detach a single property instance from any combination of entities and feature instances.

The Detach Property Property Sheet

The Detach Property property sheet allows you to accomplish the following tasks:

- Select a property instance by tagname.
- Select all property instances. You can combine All Instances with Entity or Feature. Doing so narrows your selection to the property instance that is attached to the entity or feature instance you select.
- Choose a property instance by class name.
- Select instances that meet certain conditions regarding their attributes. To specify such conditions, you must have first specified a property class. Conditional Expression only works in conjunction with Property Class.
- Select a property instance that is attached to a feature or entity. Since a property instance can be attached to multiple features or entities, you may want to combine Entity or Feature with Property Class. First choose a property class, then a Feature or Entity option. Doing so allows you to choose a property instance of a particular class attached to a particular feature or entity.

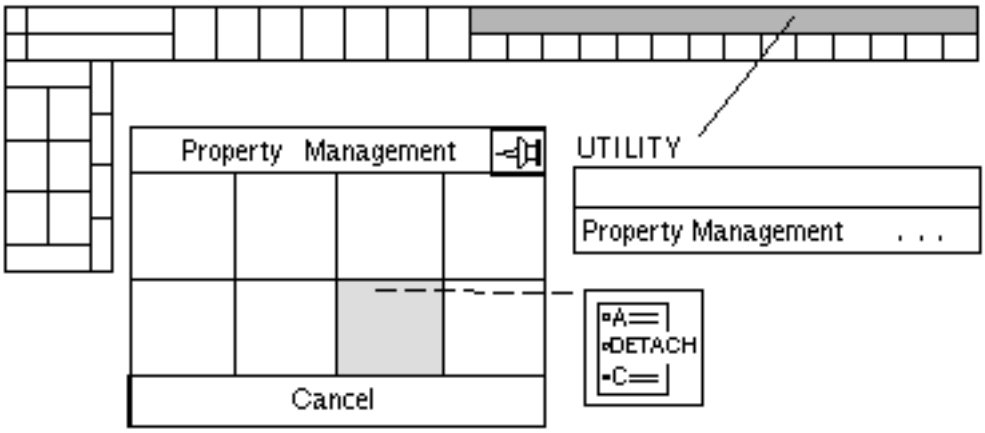


Once you choose Apply, the detachment selection options, shown in the following illustration, appear. The system then expects you to select something from which to detach.

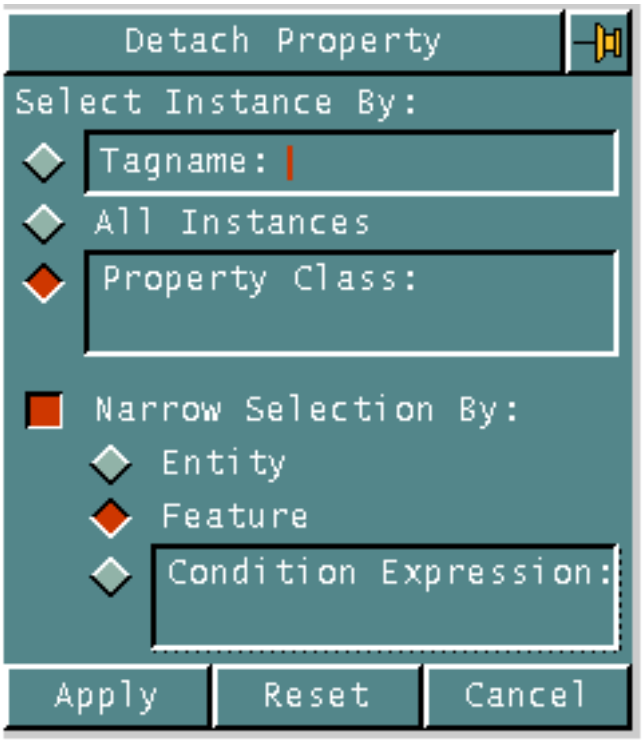
Sample Session

The following procedure uses the Detach Property option to break a reference from an existing property instance to a feature instance.

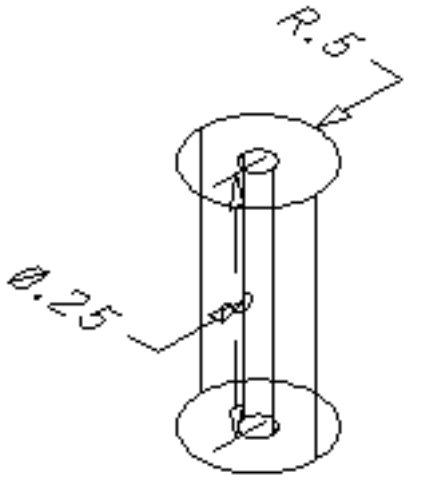
1. Choose Utility, choose Property Management and choose the Detach Property option.



2. Set the property sheet as shown, activating the property instance selection option Feature.



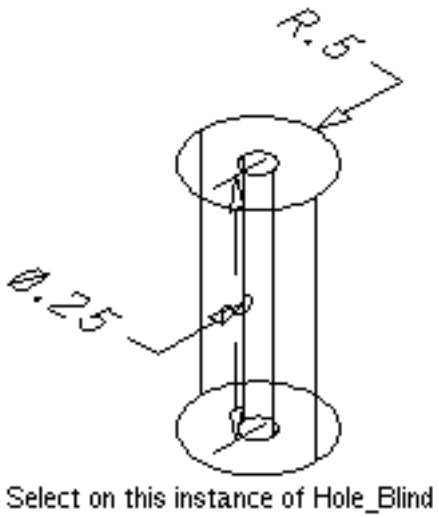
3. Choose Apply. The system now expects you to select a feature instance and also displays several detachment options.
4. Select the feature instance of Hole_Blind within the cylinder as shown. The system recognizes that an attachment to the instance of the property plating_data exists and selects that instance of plating_data.



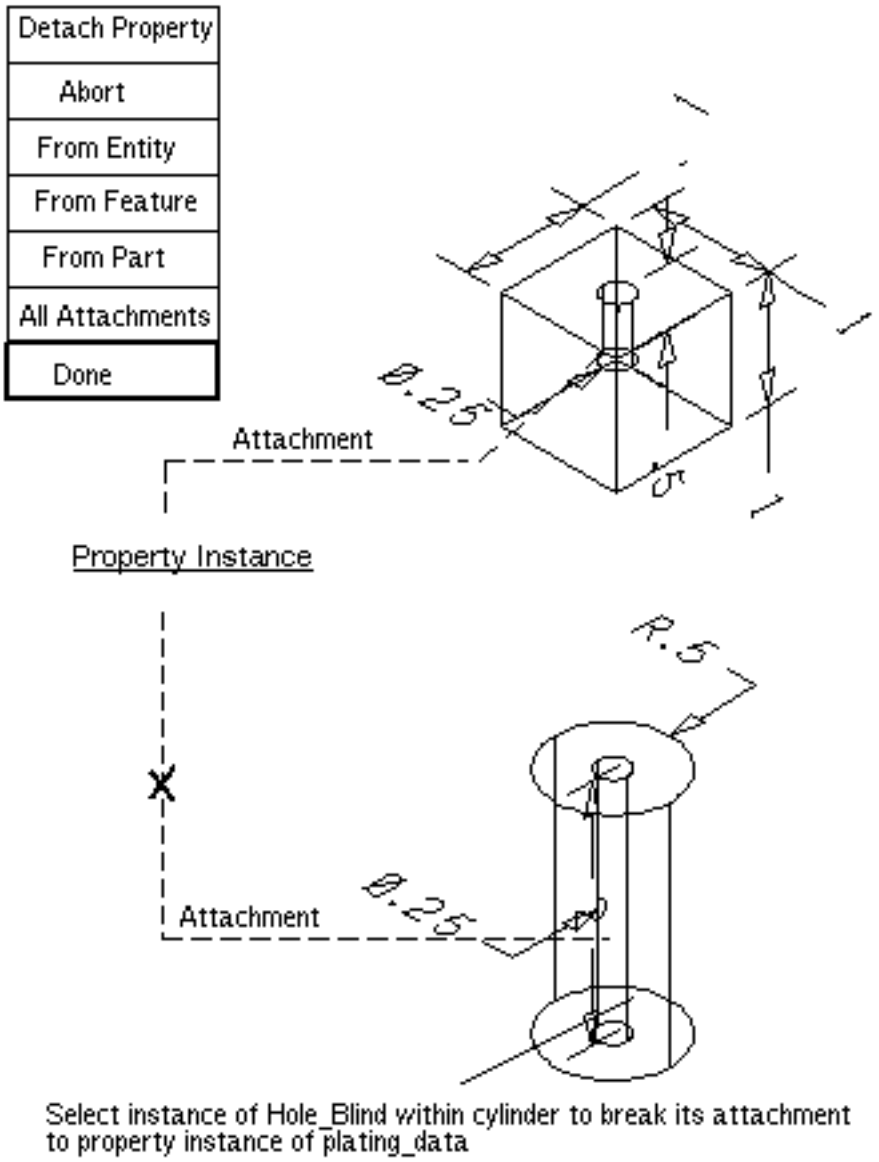
Select on this instance of Hole_Blind

5. Choose the From Feature option. The system now expects you to select a feature instance to detach.

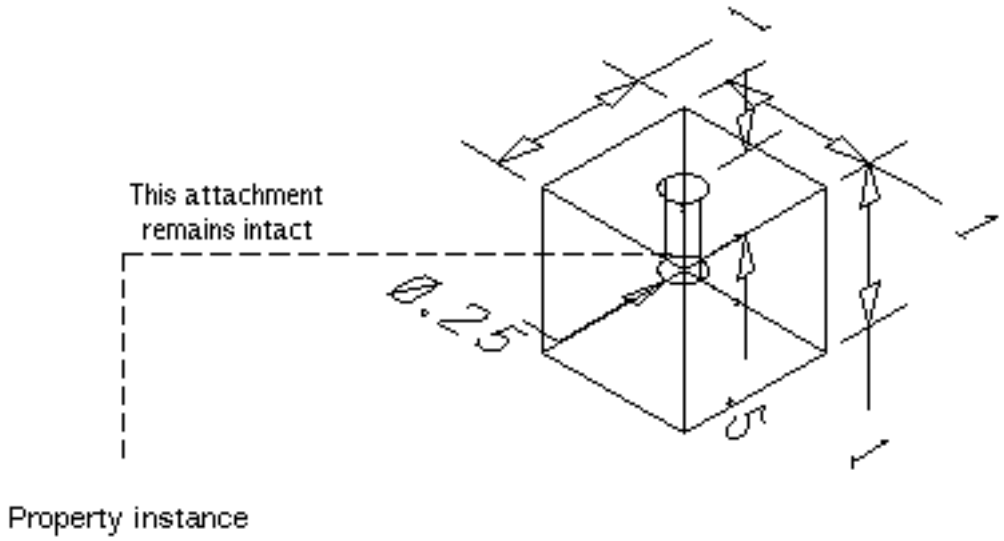
6. Again, select the instance of Hole_Blind in the cylinder.

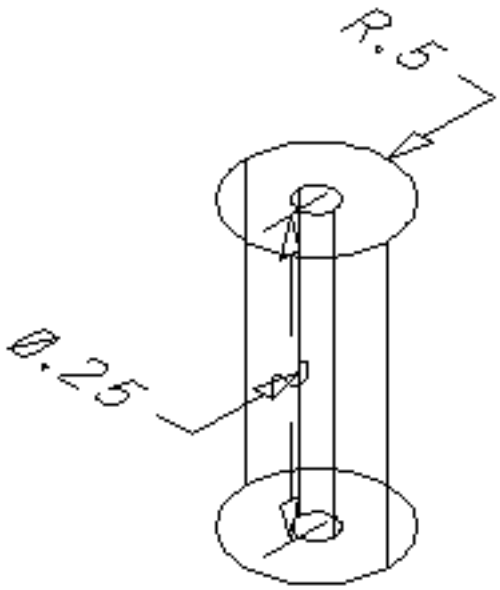


7. Choose Done. You have selected an attachment for the system to break. The system breaks that attachment and only that attachment as is shown in the following illustration.



The following illustration shows the result. The attachment to the instance of Hole_Blind in the cylinder has been broken.





Verifying a Property

The Verify Property option displays a report listing information about the property class or instance you specify. Note the following regarding this option:

- You can specify any property class within the current library. The system identifies the class name, library and version information.
- You can report on a single or multiple property instances. Use any of the property selection options.
- You can specify a brief report or a detailed report. Brief displays a minimal amount of information about a specified class or instance. Detail displays all available information, including all attachments to the selected instance.
- You can select multiple instances to verify.

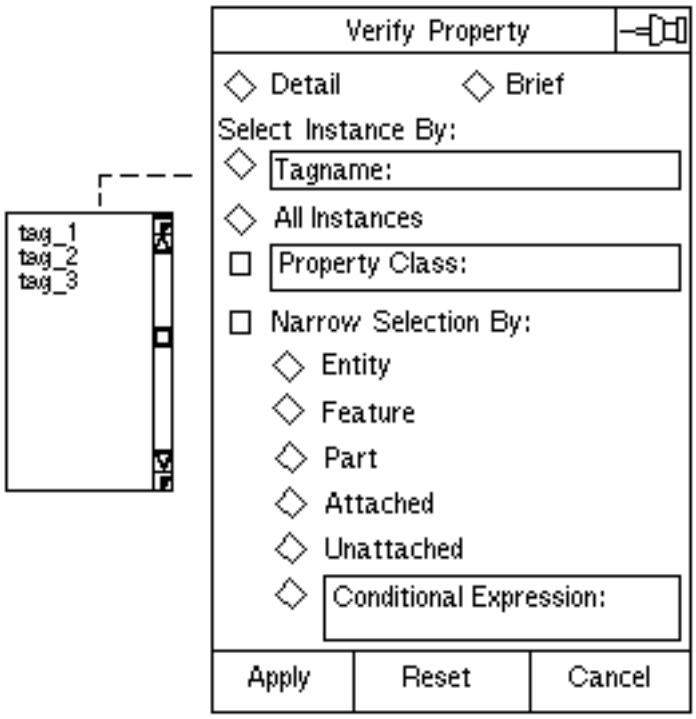
Please note: Verification of a property class requires that the LDM be active so that the system can retrieve information from the global description of that class in the current library. However, the system does not need the LDM to verify a property instance.

The Verify Property Property Sheet

The Verify Property property sheet allows you to accomplish the following tasks:

- Specify a detailed or brief report.
- Choose a property instance by tagname.
- Select all property instances. You can combine All Instances with Entity or Feature. Doing so narrows your selection to the property instance that is attached to the entity or feature instance you select.
- Select a property instance by class name.
- Select instances that meet certain conditions regarding their attributes. To specify such conditions, you must have first specified a property class. Conditional Expression only works in conjunction with Property Class.
- Select a property instance that is attached to a feature or entity. Since a property instance can be attached to multiple features or entities, you may want to combine Entity or Feature with Property Class. First choose a Property Class, then combine that selection with a Feature or Entity selection. Doing so allows you to select a property instance of a particular class attached to a particular feature or entity.

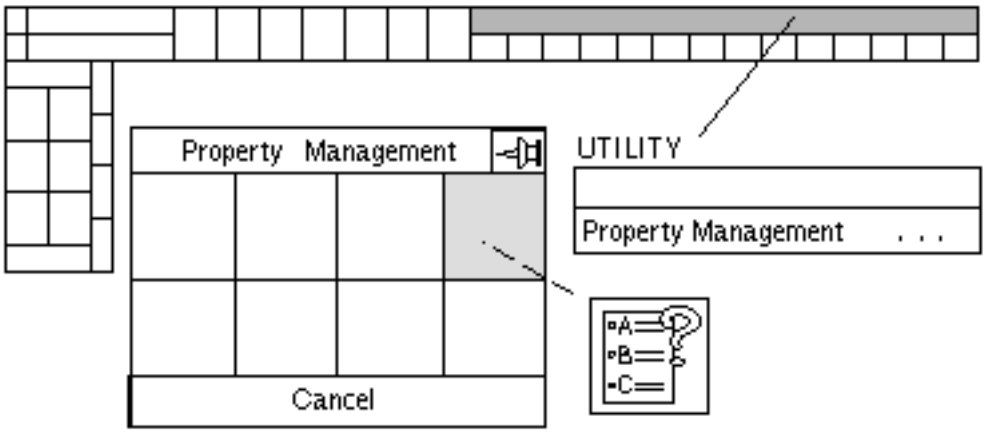
- Select all attached or all unattached instances.



Sample Session

The following procedure results in a report about the selected property instance.

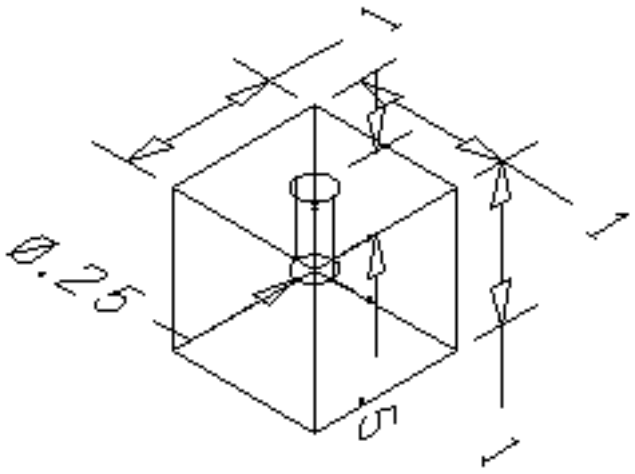
1. Choose Utility on the top bar, choose Property Management and choose the Verify Property option.



2. Set the property sheet such that Detail and Feature are active.

3. Choose Apply. The system now expects you to select a feature instance.

4. Select the instance of the feature Hole_Blind as shown.



Select on instance of Hole_Blind

5. Choose Continue. The system displays a report window.

Report Window

```
**********************************************************
class:
      class: plating_data          library: project_properties
      classtype: property          timestamp:    03-03-12:59
Static Attributes:

attributes:
     string   material                = "gold"

Attached entities:
 Hole_Through 142_I1
```

Clear | Cancel

Appendix A

Taxonomy of System Features

This appendix details the taxonomy of the system-defined features.

- Overview
- Form Features Information Model
- System Features
- Diagrams of Form Features and System Features

Overview

Advantages of Classifying Features

The term feature taxonomy refers to the way in which features are classified based on common factors such as ancestry (superclasses and subclasses) and data (attributes). There are several good reasons for providing some method for classifying features:

- The number of possible features is unlimited; no system can support every feature required by every user.
- Once features are classified, then mechanisms can be defined to support each family of features instead of having special methods for each one.
- Feature classification may lead to some common terminology.
- A taxonomy can be useful in developing data exchange standards.

System Features

System features (system-defined features) are those feature classes whose parameterizations and teach mode parts have already been defined within the system when the system is installed. All system features are stored in the system library.

Form Features Information Model

The Form Features Informational Model (FFIM) presents a conceptual model of form feature information. The FFIM is based upon a structure developed for the Product Data Exchange Standard (PDES). Please refer to Figure A-1 on page A-8.

The FFIM provides a foundation for user-defined features, common methodology and data exchange.This section describes some key terms.

Form Features

A form feature is a portion of a shape that, for the purposes of some application, is treated as an entity.

A form feature may also be classified through characteristics such as its shape, size, function, manufacturing processes, and so forth. The classifications may be narrow (counterbored, round-bottomed hole) or broad (slot).

Volume Features

A volume feature is a form feature that is viewed as a volume added to or subtracted from the previous state of the workpiece. These additive and subtractive features are described below.

Additive Features

An additive feature is a volume feature whose volume is added to the previous state of the workpiece. The value of the static attribute Operation, inherited from class Feature, is set to Add.

Additive features are protrusions, connectors or primitives, and are described below.

Protrusion: An additive feature that extends outward from the previous state of the workpiece at a single place.

Connector: An additive feature whose added volume joins the previous state of the workpiece at two distinct places. The feature may connect two separate volumes. In other cases, both connections may be with the same volume, so that the feature creates a handle on the workpiece. The instance attribute Application, inherited from class Feature, is overridden with the default value Merge so that a local merge operation is performed when the feature is inserted.

Primitive: An additive feature whose added volume does not connect to the previous state of the workpiece. The instance attributes Workpiece and Application, inherited from class Feature, are upgraded to static attributes with respective default values DB_NULLID and NoApply.

Subtractive Features

A subtractive feature is a volume feature whose volume is subtracted from the previous state of the workpiece. The value of the static attribute Operation, inherited from class Feature, is set to Subtract.

Subtractive features are depressions, passages, or voids, and are described below.

Depression: A subtractive feature whose subtracted volume emerges from the previous state of the workpiece at a single opening.

Passage: A subtractive feature whose subtracted volume emerges from the previous state of the workpiece at two distinct openings. The instance attribute Application, inherited from class Feature, is overridden with the default value Merge so that a local merge operation is performed when the feature is inserted.

Void: A subtractive feature whose subtracted volume is completely enclosed by the previous state of the workpiece.

System Features

A library of system features, named SYSLIB, is provided. This section describes those system features. The diagrams starting from page A-8 outline these features.

Tabs

A tab is a short, thin strip projecting from the edge or surface of an object. It is often used for assembly, handling, or identification purposes. A tooling tab is used for locating and clamping purposes.

Bosses

A boss is a flat surface of regular or irregular shape protruding slightly above the nominal workpiece surface and often around fastening holes to provide a seat to which other components may be attached. The surface may be formed by casting, molding, welding, and so forth. It is usually machined or ground to provide a precision surface.

Circular Bosses

A circular boss is a short, cylindrical projection from the surface of a fabricated workpiece. It is used to provide a seat for another mating part or reinforcement around a hole. The outer surface of the boss is usually machined.

Counterbores

A counterbore is a cylindrical enlargement on the end of a hole. It is often used to provide a recess for cap screw heads.

Countersinks

A countersink is a conical enlargement on the end of a hole. It is often used to receive the head of a screw or rivet.

Pockets

A pocket is a shallow non-thru-going cavity in the surface of a workpiece. It is often used for weight reduction or as a space for component parts.

Grooves

A groove is a closed circular channel on a rotational surface of a workpiece, usually for packing, seal, fluid channel, and so forth. Grooves may be found on the surface of rotational parts and along the inner wall of a bore or hole.

Flats

On an axial part, a flat is a planar surface on the workpiece, usually machined, and normal to the centerline.

Slots

A slot is a long, narrow, linear or circular channel of square or angular cross-section. Slots may:

- Be thru-slots
- Have rounded ends
- Have an undercut, as in the case of a T-slot, which is used as a mating depression for a similar protrusion configuration

Cutouts

In sheet metal, a cutout is a cut opening, regular or irregular in shape with the following characteristics:

- It has a continuous periphery (it doesn't intersect the edge of the workpiece).
- It is created by a kerf-producing process (for example sawing, flame cutting, laser cutting, and so on).

A cutout is often used as a view port, access way, or for weight reduction.

Diagrams of Form Features and System Features

Form Features Diagrams

The following diagram, Form Feature Information Model (FFIM), shows the current classification of form features. Each box in the diagram represents a feature class and its corresponding attributes. The Feature class boxes are divided into three sections:

- Class name
- Static, or class, attributes
- Attributes, or instance attributes

Features with no new attributes may be shown without an accompanying box.

System Features Diagrams

The subsequent diagrams of system features span several pages and show the taxonomy of individual system features; accompanying illustrations show how their attributes of size relate to their representations.

- Protrusions and Connectors
- Bosses and Tabs (Attributes of Size)
- Depressions I: Countersinks, Counterbores, Holes
- Depressions I: Countersinks, Counterbores, Holes (Attributes of Size)
- Depressions II: Pockets, Slots, Grooves, Flats
- Depressions II: Pockets, Slots, Grooves, Flats (Attributes of Size)
- Passages

Figure A-1 Form Feature Information Model (PDES-based)

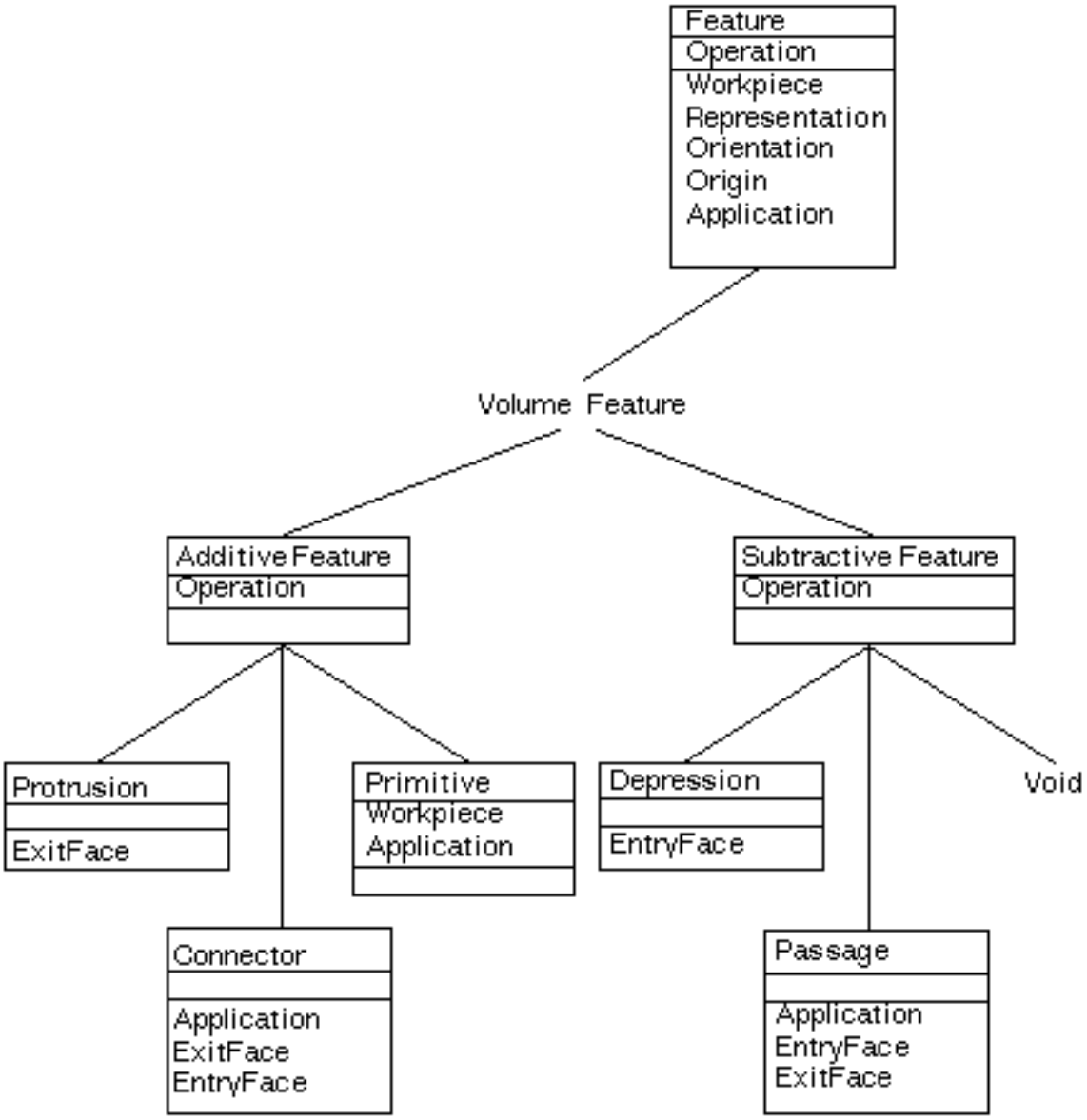


Figure A-2 System Features: Protrusions and Connectors

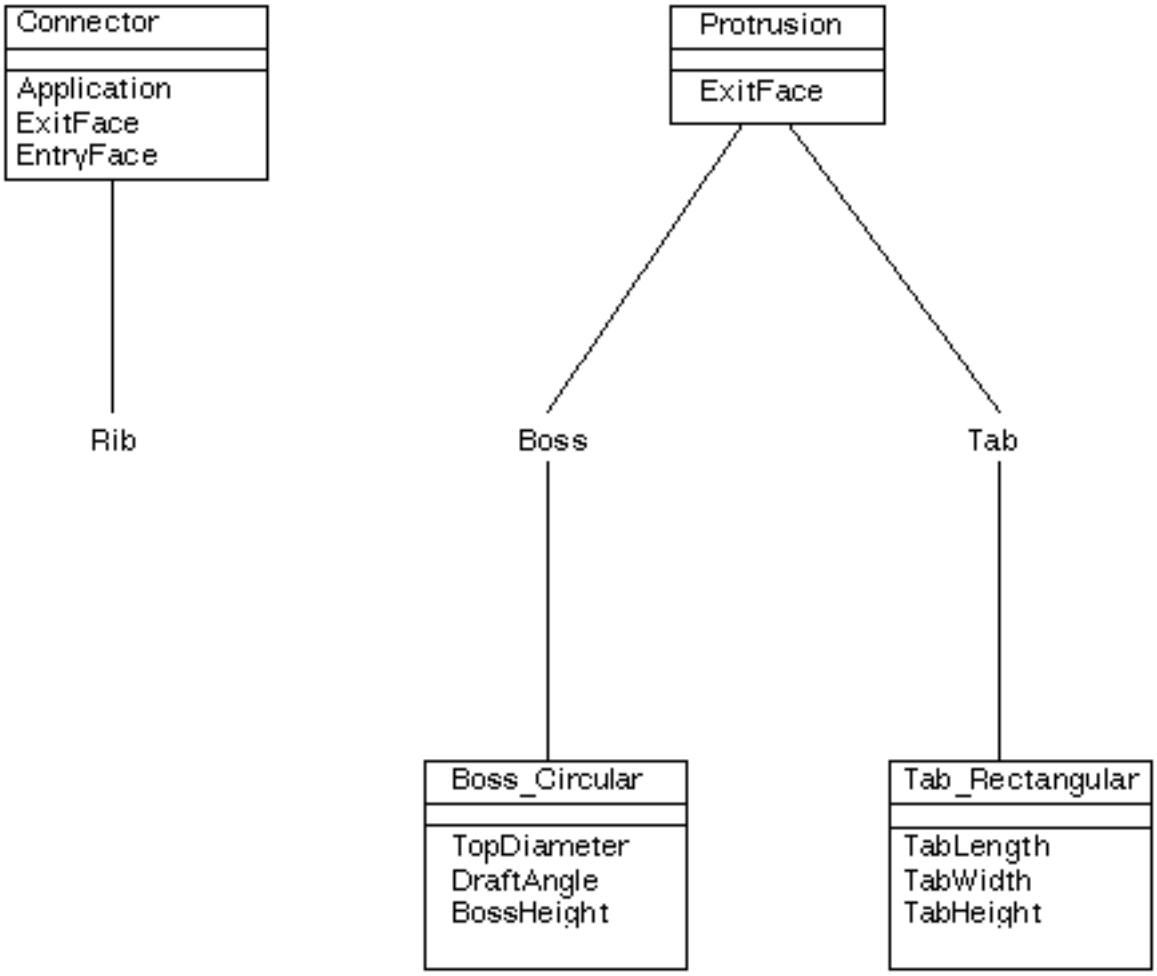
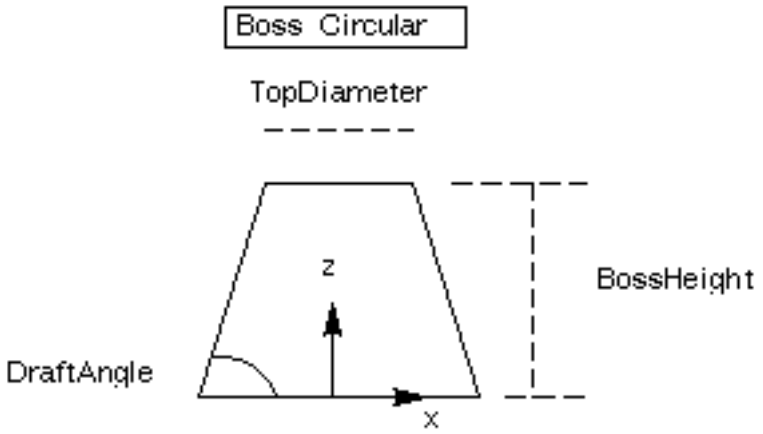


Figure A-3 System Features: Bosses and Tabs (Attributes of Size)



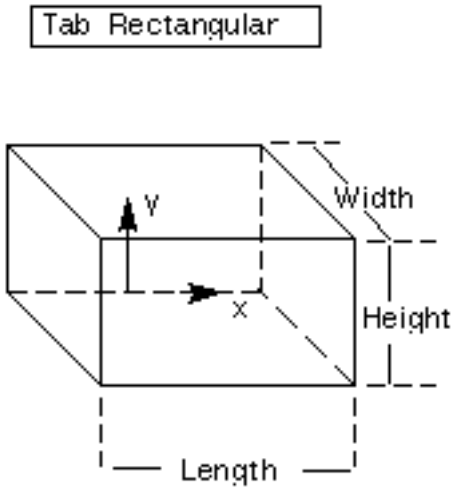


Figure A-4 System Features: Depressions I: Countersinks, Holes, Counterbores

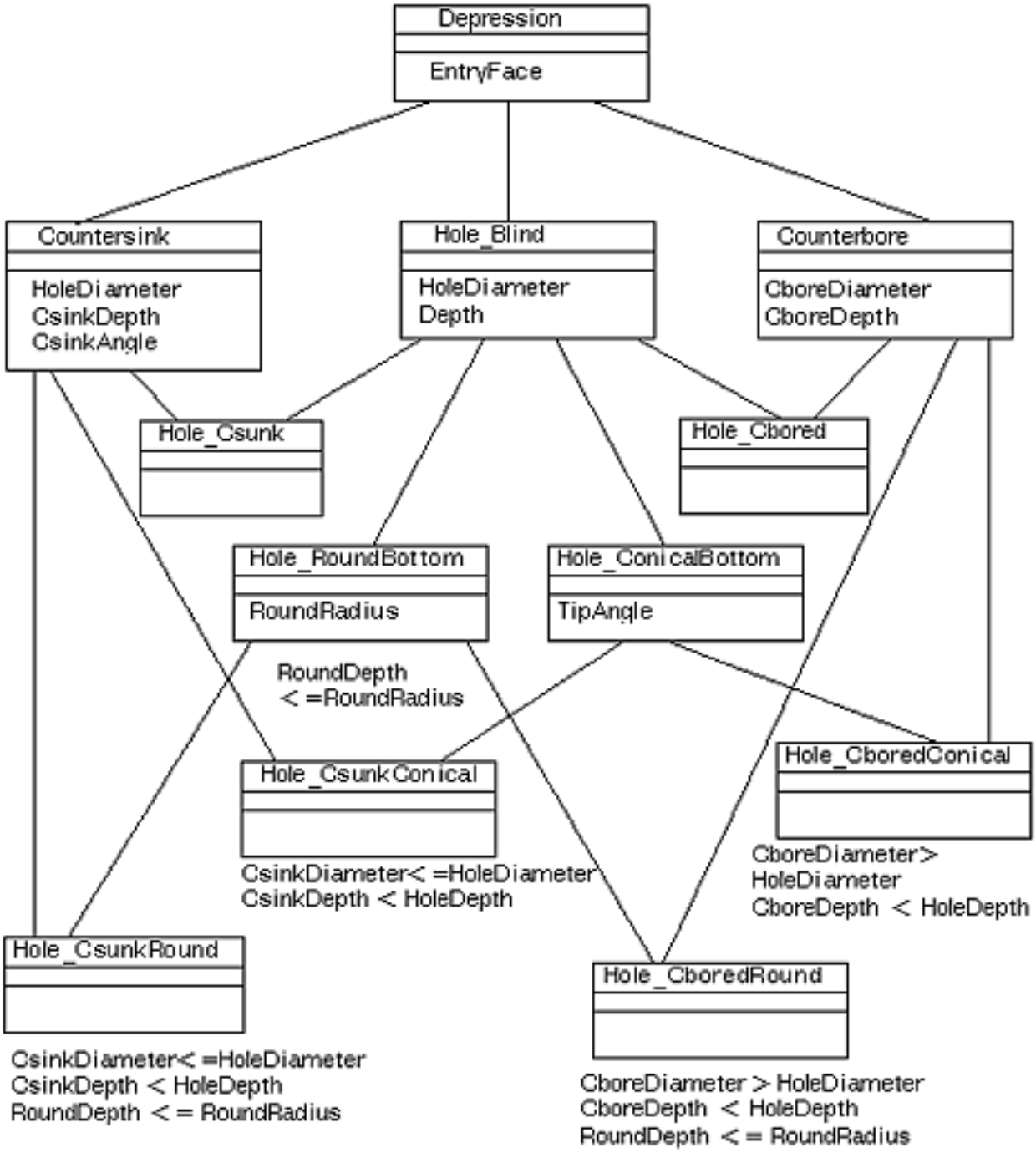


Figure A-5 System Features: Depressions I: Countersinks, Counterbores, Holes (Attributes of Size)

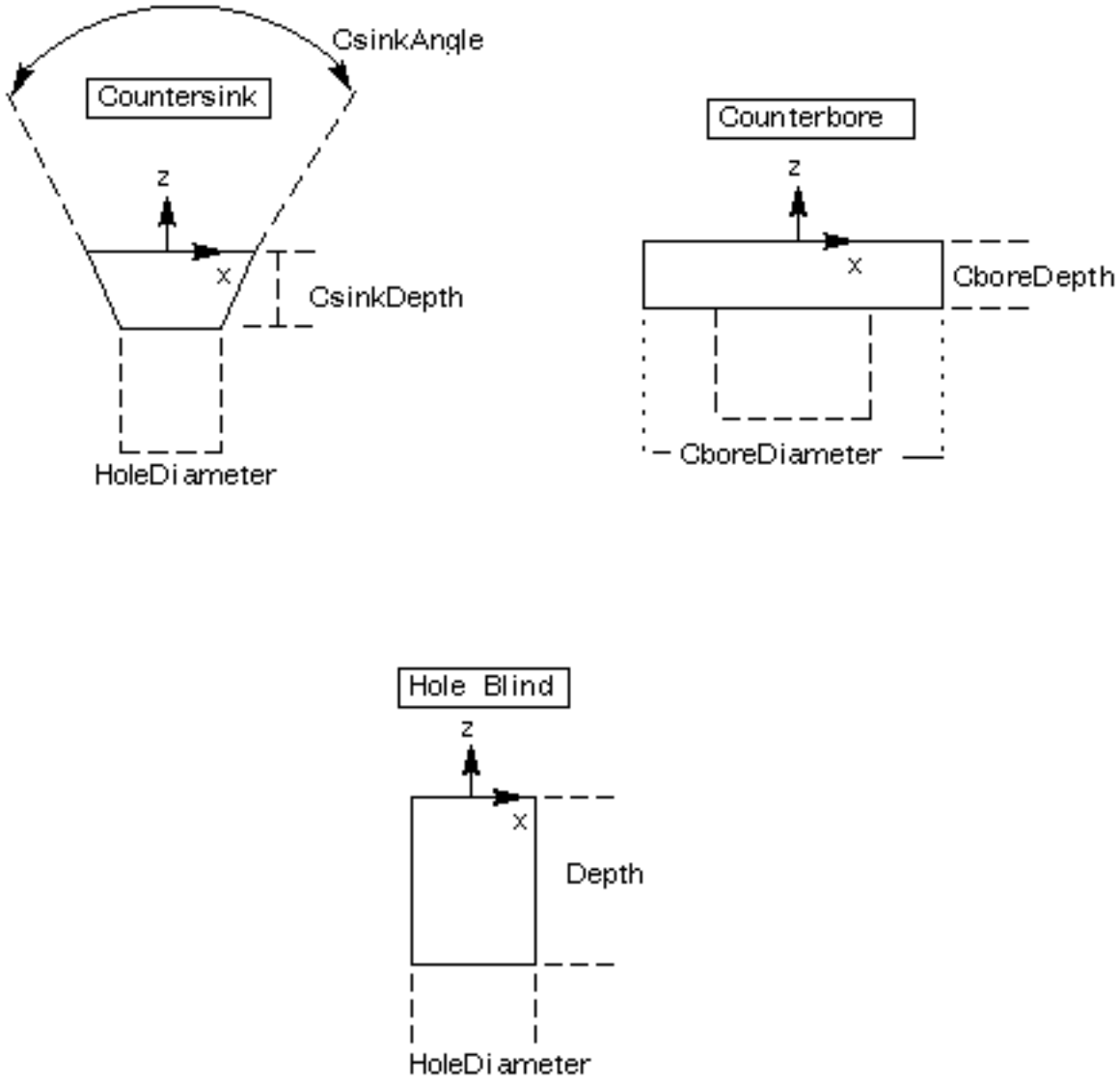


Figure A-6 System Features: Depressions I: Countersinks, Counterbores, Holes (Attributes of Size, continued)

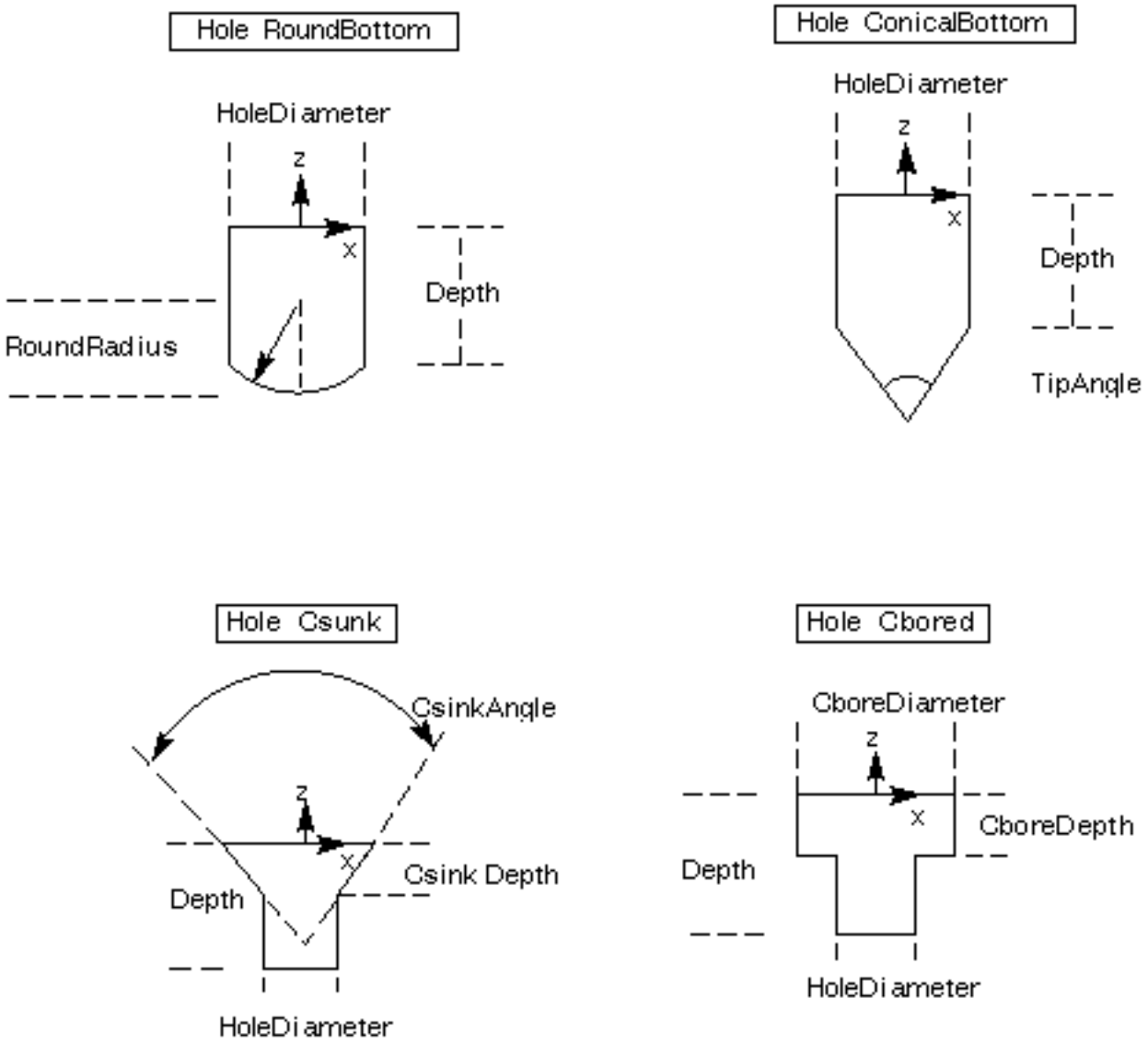


Figure A-7 System Features: Depressions I: Countersinks, Counterbores, Holes (Attributes of Size, continued)

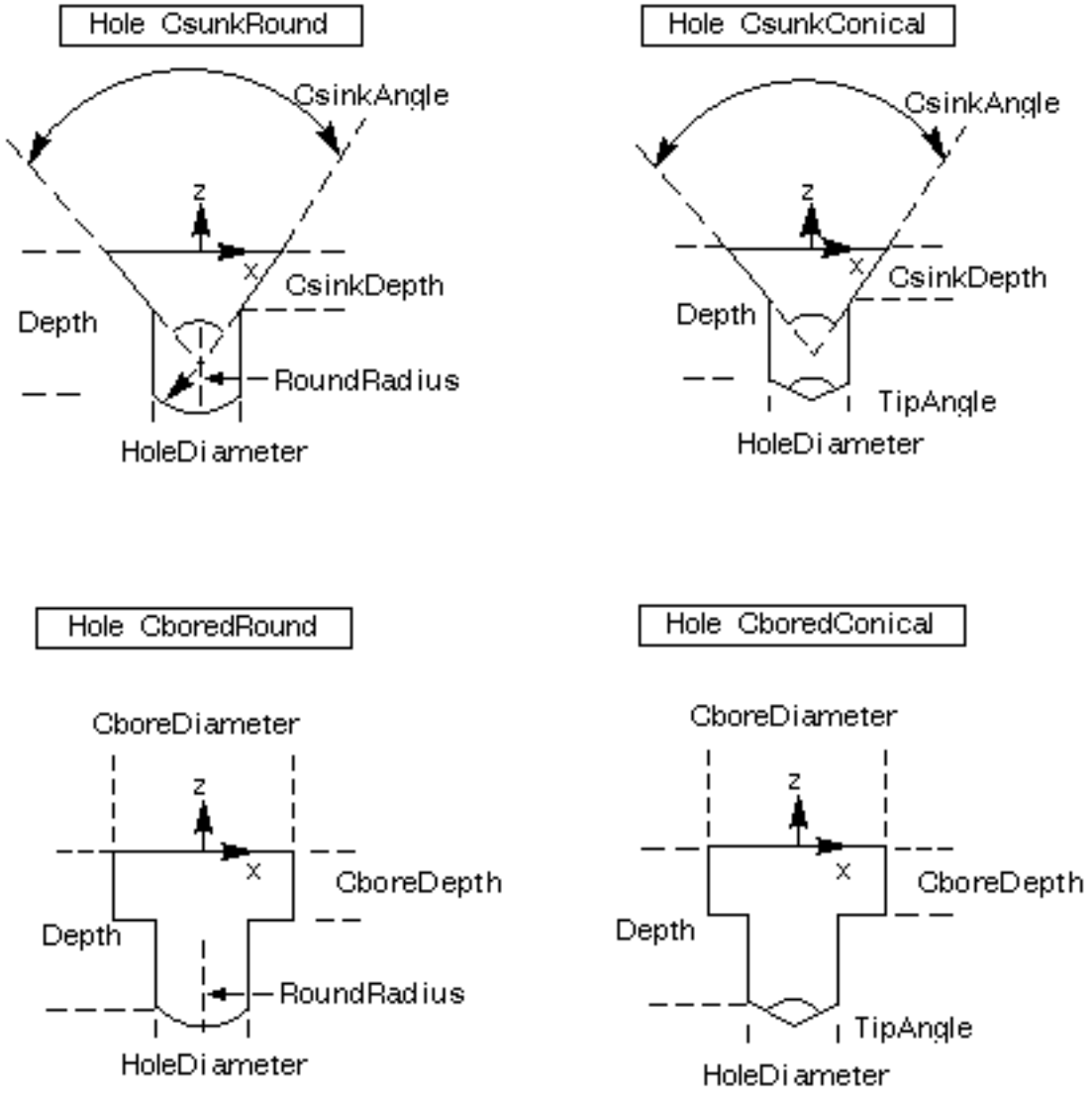


Figure A-8 System Features: Depressions II: Pockets, Slots, Grooves, Flats

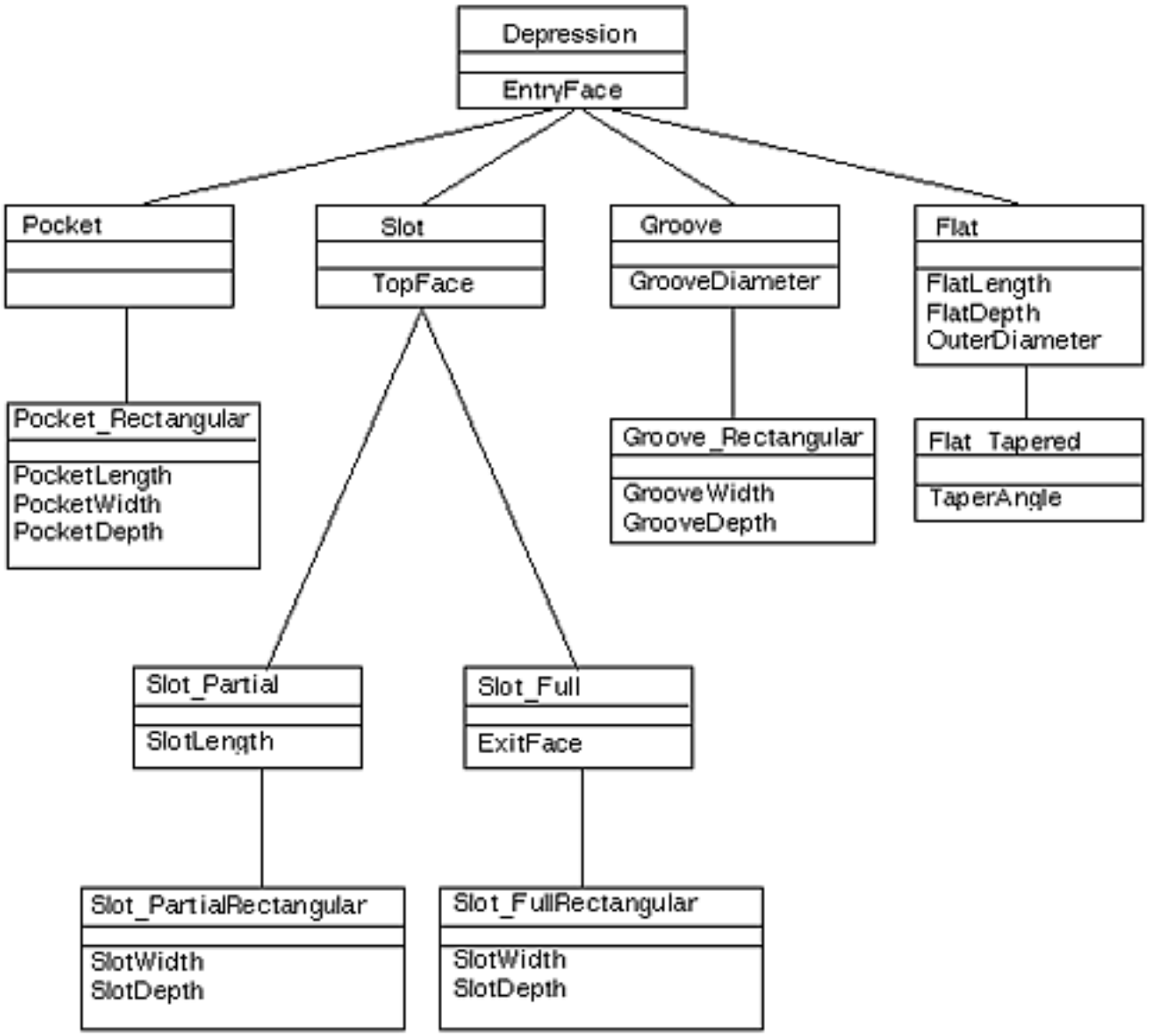


Figure A-9 System Features: Depressions II: Pockets, Slots, Grooves, Flats (Attributes of Size, continued)

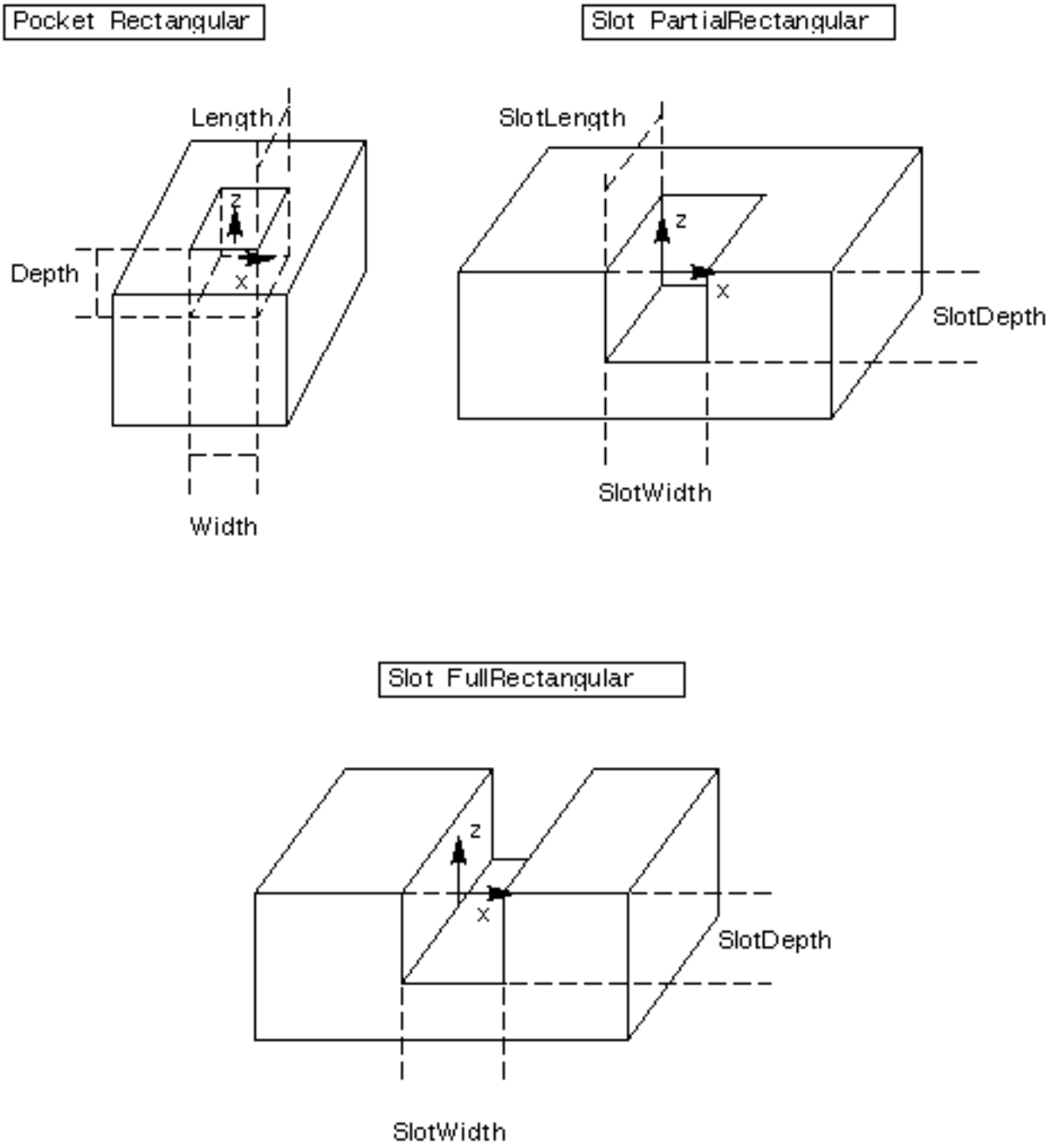


Figure A-10 System Features: Depressions II: Pockets, Slots, Grooves, Flats (Attributes of Size, continued)

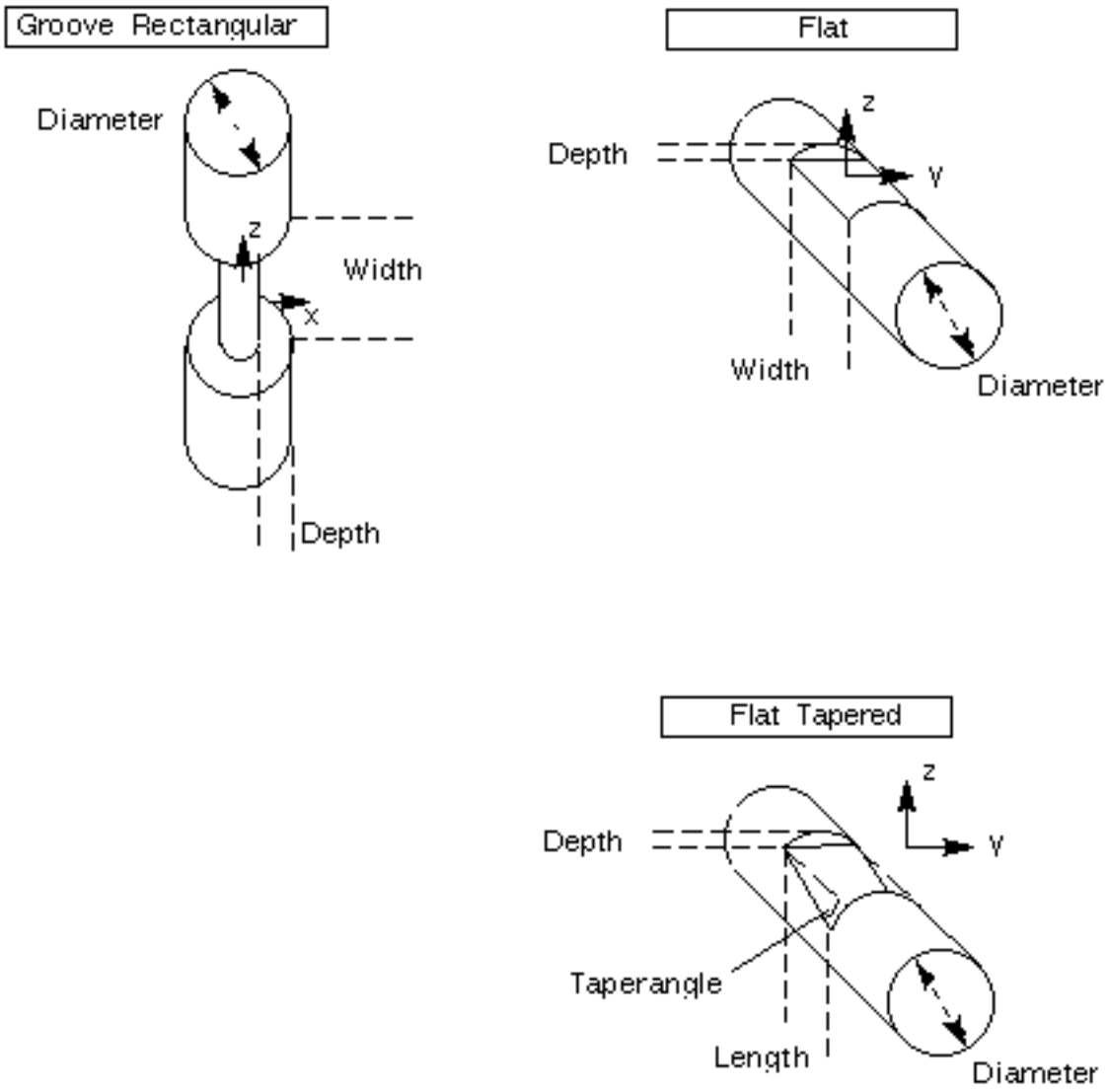
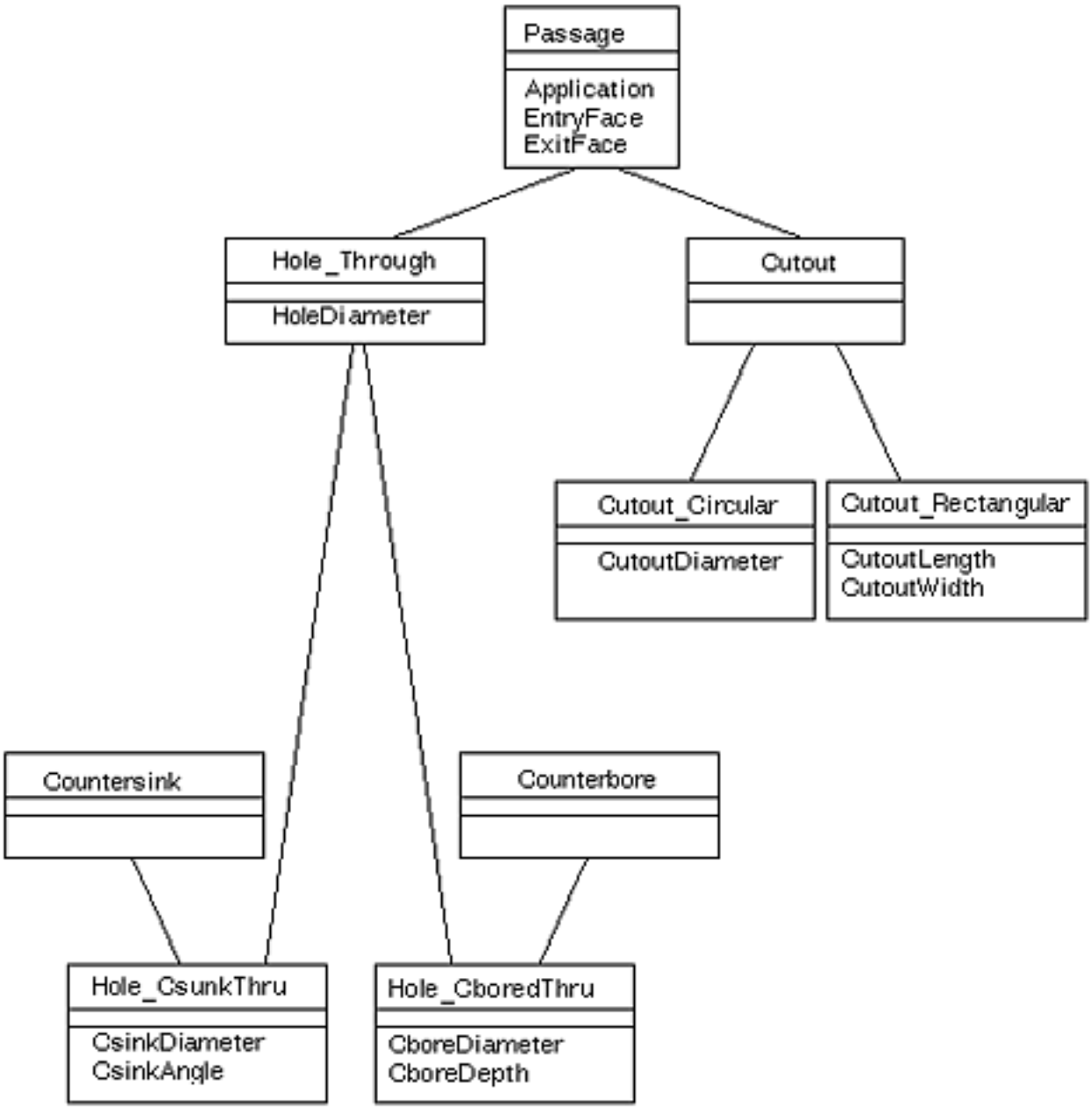


Figure A-11 System Features: Passages



Appendix B

Command File for Hole_ConicalBottom

This appendix lists the command file for the representation part of Hole_ConicalBottom which was used in the sample session in Chapter 3, “Creating a Feature.”

Inserting the Rotational Sweep

This portion of the Teach Feature command listing refers to “The Rotational Sweep” on page 3-21.

```
Select Cplane Cpname FRONT
Change View Cplane Cpname ISO
[-0.651299,0.000000,-6.049350,TOPview]
Insert Line Free Loc [0,0,0] YAngle -45.000000 Dx 1.000000 Dy
0.500000 Dx End [0.078618,0.108159,-0.186777,TOPview] End
[0.052225,0.036053,-0.088278,TOPview]
Zoom View Window [-0.039202,0.468690,-0.429488,TOPview]
[9.642208,10.906493,0.000000,DRAFTING]
[12.379871,7.241558,0.000000,DRAFTING]
Join Pcurve Chn [-0.223964,0.857314,-0.633349,TOPview] Go
Insert RotationalSweep Solid Circular
[-0.189787,0.849541,-0.659754,TOPview] Axis Y End
[-0.431208,0.861200,-0.429992,TOPview] Go
```

Inserting the User Dimensions

This portion of the Teach Feature command listing refers to “The User Dimensions” on page 3-23.

```
Change View Scale 0.500000 [-0.648902,1.121578,-0.472676,TOPview]
Zoom View Window [-0.337112,0.721060,-0.383948,TOPview]
[9.134416,10.133766,0.000000,DRAFTING]
[12.401949,7.087012,0.000000,DRAFTING]
Insert DDimension [-1.010249,1.430791,-0.420542,TOPview]
Insert Adimension Internal Lines
[-0.779073,0.778183,0.000890,TOPview]
[0.157806,0.157882,-0.315689,
TOPview] [-0.292873,0.745876,-0.453003,TOPview]
```

Associating the Attributes

This portion of the Teach Feature command listing refers to the “Association of the Attributes” on page 3-24.

Setting Constraints

This portion of the Teach Feature command listing refers to “The Constraints” on page 3-25 and “Model Regeneration” on page 3-26.

```
Associate Variable Parameter
[0.439746,-0.417188,-0.022558,TOPview] [1.238105,-0.682108,
-0.555997,TOPview]
Add Equation “HoleDiameter=2*L1”
Add Equation “TipAngle=2*A1”
Regenerate Model
```

Defining an Orientation

This portion of the Teach Feature command listing refers to the "Orientation and Origin" on page 3-26.

Exiting the Part

This portion of the Teach Feature command listing completes the sample session. Refer to page 3-15.

Appendix C

Troubleshooting

This section contains system messages and solutions that you might encounter when using feature-based modeling.

- Messages within an Application Session
- Messages for Modeling with Features
- Messages for Defining Features
- Messages for Teach Feature Definitions

Messages within an Application Session

```
Cannot find database name specified in the environment variable
LDM_LOGICAL_DB_NAME
```

Cause**:** The environment variable LDM_LOGICAL_DB_NAME specified in the `.caddsrc` does not match any of the DATA_BASE_NAME entries in the ldm_directory file specified by the environment variable LDM_DIRECTORY in the `.caddsrc` file.

Action: Check the LDM_LOGICAL_DB_NAME, DATA_BASE_NAME and LDM_DIRECTORY entries for validity, then reenter the application.

```
Cannot open the ldm_directory file as specified in the environment
variable LDM_DIRECTORY.
```

Cause: The pathname specified by the environment variable LDM_DIRECTORY in the `.caddsrc` file does not exist.

Action: Check that the pathname specified is correct, then reenter the application.

Messages for Modeling with Features

```
No face was selected for attribute %s of the feature to be inserted
with application merge.
```

Cause: The feature to be inserted requires you to specify one or more faces for correct results of the specified operation.

Action: Re-execute the command after supplying the face attributes required by the feature definition.

```
The Merge application requires a local operation but no faces were
associated in the Teach Feature session.
```

Cause: You defined class without any face attributes, or you defined class with face attributes but didn't associate any of them in teach mode.

Action: Redefine class or reteach rep and associate face attributes.

```
Error - Invalid entry - FF_ITEM_TYPE_LOCATION expected
```

Cause: The modifier Origin was specified but no location was specified.

Action: Specify a location for the Origin modifier and re-execute the command.

```
Error - Invalid entry - FF_ITEM_TYPE_WORKPIECE expected
```

Cause: The modifier Workpiece was specified but a workpiece was not specified.

Action: If the Workpiece modifier is required, specify a workpiece when re-executing the command.

```
Error - Invalid entry - FF_ITEM_TYPE_CPLANE expected
```

Cause: The modifier Orientation was specified but a Cplane was not specified.

Action: Specify a Cplane when using the Orientation modifier.

```
Error - Invalid entry - FF_ITEM_TYPE_FACE expected
```

Cause: A face modifier was specified but a face was not selected.

Action: When specifying a face modifier, select a valid face when re-executing the command.

Messages for Defining Features

```
Select a user library to define a new feature
```

Cause: You are not allowed to define a feature in the SYSLIB.

Action: Select a library other than the SYSLIB in which to define a new feature in. If another library does not exist, use Add Library to create one.

```
Feature class <NAME> is not defined in any of the accessible
libraries.
```

Cause: The feature to be created does not exist.

Action: No action is required. This message tells you that this is a new feature definition as opposed to a redefinition of an existing feature.

```
SYNTAX ERROR --- Name Conflict Naming conflict among class
attribute <name> in the new class.
```

Cause: Class attribute name is already used in the feature definition.

Action: Rename the class attribute name and re-execute the command.

```
WARNING --- Superseded The Class attribute <name>, of class
Feature, supersedes the same-named class attribute in secondary
superclass test.
```

Cause: The feature definition contains an attribute with the same name and type as of an inherited attribute. The class attribute overrides the inherited superclass attribute.

Action: No action is required. This is a warning to inform you that an inherited attribute is being overridden.

```
Library <name> already exists.
```

Cause: The library name specified in the Add Library command already exists in the database.

Action: Enter a different library name and re-execute the command.

```
Cannot delete library <name>. Feature definitions still exist in
it.
```

Cause: Feature definitions still exist in the library specified.

Action: Verify that you have chosen the proper library to be deleted. Delete all feature definitions in the library, or specify the Contents modifier of the Delete Library command.

```
Feature class <name> has subclasses and cannot be deleted.
```

Cause: The feature class is a superclass of other feature definitions and therefore cannot be deleted until all subclasses have been deleted.

Action: Delete all subclasses prior to deleting a superclass.

Messages for Teach Feature Definitions

```
WARNING - Geometry attribute <name> is not associated.
```

Cause: The geometry attribute specified in the feature definition was not assigned a necessary geometric entity.

Action: No action is required. However, you should recheck the teach part to verify whether the geometric attribute should indeed be associated. If the attribute should be associated, you must enter Teach mode and use Associate Attribute to create the association.

Appendix D

Setup Procedures

This appendix describes procedures for various tasks associated with a features database.

- Installing and Setting Up the Features Database
- Accessing a Local Features Database
- Switching Features Databases
- Copying a Features Library
- Backing Up Features Libraries
- Updating the LDM with a New Revision

Installing and Setting Up the Features Database

Please note: When you installed your application package(s) with SLIC, a local features database and server process were automatically installed using the defaults specified by `/usr/apl/cadds/scripts/ldmqload`. This setup is sufficient for the majority of users. However, if you need to have a nonstandard features installation or the installation of features failed during install, read on.

The Features Database

The features database holds features libraries, definitions, and representations. The system administrator installs this once. A features database is identified by its name and the directory in which it resides.

Please note: The features database was formerly called the Local Data Manager or Local Process Controller. Many names of associated files, scripts, environment variables, and error messages still contain the acronyms ldm, LDM, or lpc.

Features Process

The features process (LDMSERV) allows multi-user local or remote access to the features database. The features process must be running for you to use features. Every user should know how to start the features process. This process must be started in the following cases:

- Each time the system is rebooted
- Each time a new software version is installed
- Automatically when the features database is installed

Your configuration file is called either `.caddsrc`, `.caddsrc-local`, or `.pmodelrc`. This document refers to the application configuration file as the `.caddsrc-local` file. If you are using a `.caddsrc` or `.pmodelrc` file, make the appropriate substitution in these procedures.

Features Libraries

The features libraries contain the features themselves. There is a single system-supplied features library, called SYSLIB. In addition, you can create multiple user-defined features libraries. In our example, the features libraries are located in the directory `/usr/system-name/features`. For example, if the hostname of the system is vega, this directory would be `/usr/vega/features`.

You can put your features libraries into another directory, but we recommend this naming structure because it is easily recognized and can be shared among several systems.

The ldm_directory file, located under `/usr/ldm`, is a configuration file that defines one or more features databases. Each features database definition contains the following:

- Name of the features database and the host system
- Path name of the features database directory
- Ethernet port for process communication

Each line consists of variable = value. Lines can appear in any order. Spaces and tabs are ignored. Blank lines are not allowed. Lines with semicolon (;), pound sign (#), or slash-asterisk (/*) in the first column are treated as comments and are ignored.

Each features database definition consists of three lines as follows:

```
features_db: HOSTNAME = vega
        DATA_BASE_NAME= features_db:/usr/ldm/lpc_db
        ETHERNET_PORT = 1051
```

The entries are:

features_db	Name of the features database. This name is used in the command that starts the features process, as `ldmserv features_db start`.
HOSTNAME	Name of the system on which the features process is running. If the features database is local, it is the name of this system.
DATA_BASE_NAME	Name of the features database followed by the full path name of the database files. (Note that this database name is set in the `.caddsrc-local` file as the variable LDM_LOGICAL_DB_NAME so that the software knows which features database you want to use.)
ETHERNET_PORT	Used for communication between software executables and the features process. Use 1051 for your first features database, 1052 for the next features database and so on.

Creating a Local Features Database Manually

A local features database means that the features database and the features libraries are both on your local system.

Setting up a local features database has two main actions:

- Install the local features database
- Test the features database setup

Installing the Local Features Database

To create a local features database, you must be root user. Follow the procedure below to create a local features database:

1. Log in as root user.

   ```
   login: root
   password: {enter password if system secure}
   ```

2. The home directory for any user should contain a `.caddsrc-local` file. If it doesn't, create it.

   Ensure that the following variables are in the `.caddsrc-local` file. The values shown here are recommended. They specify that a features database called features_db is to be installed in the directory /usr/ldm. If you want to use a different directory or a different database name, make the appropriate substitutions in these lines and in this procedure.

```
For all platforms:
setenv FBM_SYSLIB_PATH /usr/hostname/features)

setenv LDM_LOGICAL_DB_NAME features_db
setenv LDM_DIRECTORY /usr/ldm/ldm_directory
setenv LDM_DIRECTORY_SERVER /usr/ldm/ldm_directory
```

   For more information about these variables (and other required features database variables), see the later section Features Variables in the `.caddsrc-local` file.

3. The features database directory, `/usr/ldm`, takes .5 MB of disk space when it is installed. This space grows as features are added. We recommend that you start with about 10 MB of space. If you do not have enough space in the /usr partition, create a link to another partition which has sufficient space, such as /usr2.

   ```
   # mkdir /usr/ldm
   ```

   Or

```
# mkdir -p /usr2/ldm
# chmod -R 777 /usr2/ldm
# ln -s /usr2/ldm /usr/ldm
```

4. The features libraries, in /usr/system-name/features, holds both the system features libraries (about 2MB) and user features libraries. We recommend that this directory also have 10 MB available. If you do not have enough space in the /usr partition, create a link to another partition which has sufficient space, such as /usr2.

```
# mkdir -p /usr/system-name/features
```

   Or

```
# mkdir -p /usr2/system-name/features
# ln -s /usr2/system-name/features /usr/system-name/features
```

5. Run the `ldminstall` command with values from the user's `.caddsrc-local` file. This installs the features system library and the features database and starts the features process.

```
# /usr/apl/cadds/scripts/ldminstall  -rcpath ~username
```

6. Check the status variable to see if `ldminstall` ran correctly:

```
# echo $status
0
```

   If the environment variable $status is zero, then the new database has been created and the server process has been started. If not, then an appropriate error message will be printed. See the section Starting the Features Process for troubleshooting information. Type `ldminstall -help` for online information.

7. Check that your new database has been created. List the /usr/ldm directory and look for the files `lpc_dbE, lpc_dbI, and lpc_dbT`.

```
#ls /usr/ldm
```

8. Check the /usr/system-name/features directory to see that the system library, FF-LIB-SYSLIB, is there.

```
#ls /usr/ldm/lpc_db*
lpc_dbE lpc_dbI lpc_dbT
#ls /usr/system-name/features
FF-LIB-SYSLIB
```

Testing the Features Database Setup

To ensure that your features database is set up correctly:

1. Start CADDS.
2. When the Local Data Manager screen appears, the upper-left menu displays LDM MODE: PARTS. Click on LDM MODE to display LDM mode options. Click on FEATURES to bring up the Features page.

   If the application does not appear, you may see the message
   `LDM is not up.`

   Recheck the setup procedure and see the section called Starting the Features Process.

Creating a Shared Features Database

Instead of setting up a features database on each system, you can set up a local database on one system and access it remotely from other systems.

Advantages of a Shared Database

The advantages of setting up a shared database are

- The features database needs room on only one system.
- You can share the user-defined features libraries. This saves space and gives users access to each other's features.
- If users do not have their own designated systems, several systems can be set up identically and users can easily sit at any system and be assured of correct access.

Setting up a shared features database has two main steps:

1. Set up the client systems
2. Test the features database setup

Set up the Client Systems

As root user, perform the following steps on each client system. Note that steps 3, 4, and 5 differ for different platforms.

1. Log in as root user.

   `login:` root
   `password:` {enter password if system secure}

   The ldm_directory file contains the following:

   features_db: HOSTNAME = system-name
   DATA_BASE_NAME= features_db:/usr/ldm/lpc_db
   ETHERNET_PORT = 1051

2. Create a /usr/ldm directory. Copy the ldm_directory file from the remote host:

```
# mkdir /usr/ldm
# cd /usr/ldm
# rcp system-name:/usr/ldm/ldm_directory .
```

3. Edit the /etc/fstab directory to mount the features library from the host system.

   For SunOS , Digital UNIX , IRIX:

   Activate the new entry with the `mount -a` command.

   When you are mounting a features library directory from a remote system, the directory on the local system must be exactly the same as on the host system. In the sample below, system vega's directory /usr/vega/features must be called /usr/vega/features on the local system.

```
# vi /etc/fstab
insert: vega: /usr/vega/features /usr/vega/features nfs
rw,bg,intr,hard 0 0
# mount -a
```

Please note: Make sure you set the proper privileges in the host system's `/etc/exports` file before attempting to mount.

The above procedure shows one way that the local system can be mounted to the remote system. If the local system already has a mount to the host's /usr file system, then you need to make a link to the correct directory. See the *Network and Administration Guide* for more information on mounts.

For HP-UX :

Use SAM utility to mount /usr/system-name/features
or
Edit the `/etc/checklist` file and add the following line:

system-name:`/usr/`system-name`/features /usr/`system-name`/features nfs rw,bg,hard,intr 0 0`

For Solaris:

Edit the `/etc/vfstab` file and add the following line:

system-name:`/usr/`system-name`/features - /usr/`system-name`/features nfs - yes - rw,hard,bg`

Then issue the UNIX command: `mountall`

For AIX:

Use SMIT utility to mount NFS file system locally.

4. Edit each user's `.caddsrc-local` file. Insert the name of the remote host in place of '/bin/hostname' in the FBM_SYSLIB_PATH variable.

```
setenv FBM_SYSLIB_PATH /usr/system-name/features
setenv LDM_LOGICAL_DB_NAME features_db
setenv LDM_DIRECTORY /usr/ldm/ldm_directory
setenv LDM_DIRECTORY_SERVER /usr/ldm/ldm_directory
```

   For an explanation of these variables (and other required features database variables), see the section on features variables in *Managing CADDS 5*.

5. Be sure that the LDMSERV process is running on the remote system, here named vega for example:

   For SunOS

```
# rsh vega ps -ax | grep LDMSERV
2636 p3 S 0:00  LDMSERV features_db
```

   For HP-UX

```
#remsh vega ps -ef | grep LDMSERV
2636 p3 S 0:00  LDMSERV features_db
```

   For Solaris, IRIX, Digital UNIX, and AIX

   # rsh vega ps -ef | grep LDMSERV

Please note: You cannot start a remote LDMSERV process on the local system.

6. If the application is already running on the local system, exit and restart it. The system reads the changed `.caddsrc-local` file and activates the new environment variables.

Test the Features Database Setup

To ensure that your features database is set up correctly:

1. Start CADDS.
2. When the Local Data Manager screen appears, the upper-left menu displays LDM MODE: PARTS. Select LDM MODE to display LDM mode options. Select FEATURES to bring up the Features page.

If the application does not appear, you may see the message:

```
LDM is not up.
```

Recheck the setup procedure and see the section called Starting the Features Process.

The message `Library not accessible` means that your remote system cannot find the features library. If you get this message, check your mounts against the output of the LIST LIBRARY command.

If you are sharing a features library with other systems, be sure to use the correct shared path when you use the ADD LIBRARY command so that others can see the new library. In our example, always use the path /usr/system-name/features when you add a library.

Features Libraries on Client Systems

The configuration setup in the previous section, Creating a Shared Features Database, describes a setup with the features database and the features libraries on the same host system. You also can put the features libraries on several different systems and share access.

You might want to do this to maximize disk space or decrease network traffic. For instance, to decrease network traffic, you might want frequently used features libraries local to your system even if you are remotely accessing the features database. This flexibility requires a cross mounting scheme.

Example of Cross Mounting

As an example, we use three systems, sys1, sys2, and sys3. System sys1 has the features database on it and the shared system library. The `FBM_SYSLIB_PATH` variable on all three systems is `/usr/sys1/features`.

Each of the three systems has a mount to the location of each of the other two system's user-defined features libraries. For instance, system sys1 has the following lines in its `/etc/fstab` file:

```
sys2:/usr/sys2/features /usr/sys2/features nfs rw,bg,hard,intr 0 0
sys3:/usr/sys3/features /usr/sys3/features nfs rw,bg,hard,intr 0 0
```

Systems sys2 and sys3 have corresponding entries for the other two systems.

All three systems can access the features on each of the systems. Users can choose to which of these directories they will add new features or new features libraries.

See the *Network and Administration Guide* for more information on mounts.

Starting the Features Process

To use features, the features process must be running on the features database system. When you install a local database, the `ldminstall` command starts the features process automatically.

The server does not need to be restarted with each session. You need to restart the server in the following situations:

- If the features process was stopped or was killed
- If the host system was rebooted
- If a new revision of the application was distributed

The `ldmserv` command can start and stop the features process. The procedure below shows how to start the process. See `ldmserv -help` for online information.

Run the command as a regular user, not as root user. If you are currently root user, switch to being a regular user, using `su -`. The hyphen with the `su` command sources the `.cshrc` and `.login` files for the user.

In the example below, features_db is the name of the default features database, located in the /usr/ldm directory. If the database name in the directory file is different from features_db, then use that name instead.

- To start the process as a regular user:

  For SunOS

```
% ldmserv features_db start
features_db started
```

  For Solaris 1.1.1B/1.1.2

```
% ps -ax | grep LDMSERV
  2636 p3 S 0:00 LDMSERV features_db
```

  For HP-UX, Solaris, Digital UNIX, IRIX, and AIX

```
% ps -ef | grep LDMSERV
  2636 p3 S 0:00 LDMSERV features_db
```

- To start the process as root user:

```
# cd /usr/apl/cadds/bin
# /usr/apl/cadds/scripts/ldmserv  -rcpath
      /usr/apl/cadds/scripts/templates
```

A message verifies that the features process is running. Always check for this confirmation message.

If the Process Does Not Start

If you are having trouble starting the features process, look into the directory /usr/ldm/server_output and read the log files there. These log files are named `features_db###`. The string of numbers is the date and time. You may discover the cause of your problems. You can delete these files periodically when they are no longer useful.

See also the troubleshooting information in Appendix C, “Troubleshooting.”

When you start the features process, temporary database files with names matching L*E**,** L*I**,** and L*T appear in the /usr/ldm directory. If the features process does not start, and these files exist (from a previous process), do the following:

```
# ldmserv features_db stop
# ls /usr/ldm
```

If these files are still listed, as shown below, then execute the `rm` command and try to start the features process again:

```
L*E
L*I
L*T
# rm L*
# ldmserv features_db start
```

Warning

Do not delete these temporary database files without first running **ldmserv features_db stop**.

Releasing the Port

If you do delete the temporary database files, the `stop` command may not release the Ethernet port. Starting the process produces an Ethernet port error. Kill the process with `kill -9` process-number, then restart it. Do not edit the Ethernet value in this case, as it invalidates the port value for all users of the database process.

Choosing a New Port

If the features process does not start and you receive the following message, choose another port:

The configured ETHERNET_PORT is being used already by another process.

Choose a port number above 1051. To select a port number, check for ports in use with the following command, where system_name is the name of the system you are working on.

SunOS and HP

`/usr/etc/rpcinfo -p` <system_name>

Solaris

`/usr/bin/rpcinfo -p` <system_name>

Select a port value that is not listed by `rpcinfo -p`, but note that not all network services are shown by `rpcinfo`. For instance, any features processes that are running are not listed!

Accessing a Local Features Database

If you are a features user and your system administrator has set up a features database on your system, but your user login is not yet set up, use this procedure.

1. The following lines should be in your `.caddsrc` file. These show the recommended values. The features database is called features_db and is in the directory /usr/ldm. The features database is on the system you are logged into.

```
setenv FBM_SYSLIB_PATH /usr/`/bin/hostname`/features
setenv LDM_LOGICAL_DB_NAME features_db
setenv LDM_DIRECTORY /usr/ldm/ldm_directory
setenv LDM_DIRECTORY_SERVER /usr/ldm/ldm_directory
```

2. Check to see whether the features server process is already running.

```
% ps -ax | grep LDMSERV
2636 p3 S  0:00  LDMSERV features_db
%
```

Please note: For the HP platform use **-**ef instead of -ax.

If it is not running, start it and check that it is running.

```
% ldmserv features_db start
features_db started
% ps -ax | grep LDMSERV
2636 p3  S  0:00  LDMSERV features_db
```

3. Check that the features database is correctly accessed.

Switching Features Databases

You may want to access a features library that is registered to a different features database than your system is set up to access. (This procedure assumes that your system name is sirius and you need to access a feature that is registered to the features database on vega.). Follow the procedure below.

1. Define the second features database in the `ldm_directory` file as follows:

```
features_db1: HOSTNAME = sirius
        DATA_BASE_NAME=features_db1:/usr/ldm/lpc_db
       ETHERNET_PORT = 1051
;
features_db2: HOSTNAME = vega
        DATA_BASE_NAME=features_db2:/usr/ldm/lpc_db
       ETHERNET_PORT = 1052
;
```

The database `features_db1` is on sirius. The database `features_db2` is on vega.

Please note: Each database name and a correspondent Ethernet_Port number must be unique.

2. Mount the relevant features library so you can access it from sirius. For instance, if the features libraries on vega are in the directory /usr/vega/features, put the following in your `/etc/fstab` file:

```
# vi /etc/fstab
insert: vega:/usr/vega/features /usr/vega/features nfs
rw,bg,hard,intr 0 0
# mkdir -p /usr/vega/features
# mount /usr/vega/features
```

3. If you are running CADDS, you must exit for the next step.
4. Edit the `~/.caddsrc` file. Change the LDM_LOGICAL_DB_NAME variable to features_db2.
5. Be sure that the features process, features_db2, is running on the remote system, here vega:

```
% rsh vega ps -ax | grep LDMSERV
2636 p3  S  0:00  LDMSERV features_db2
```

Please note: For the HP platform use -ef instead of -ax.

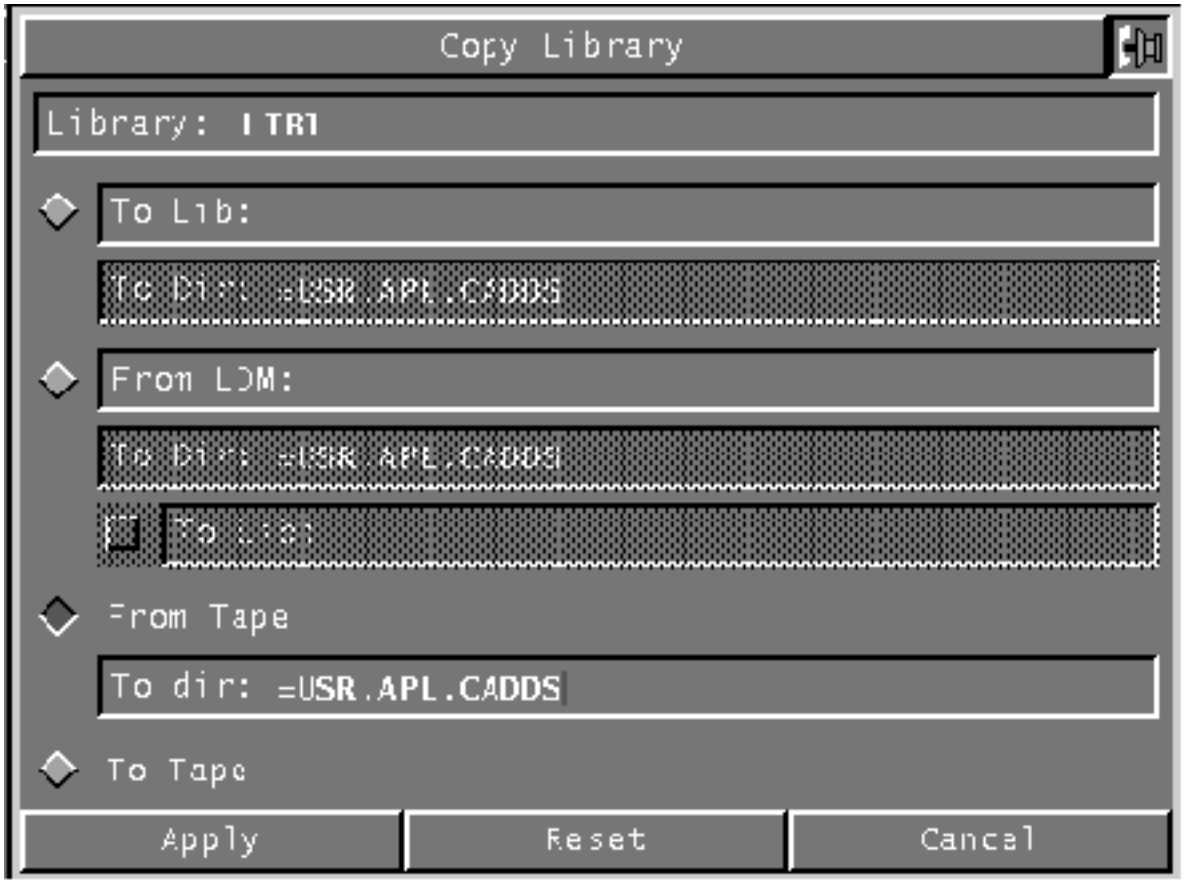
6. Reenter `cadds`.

```
% cadds
```

You are now accessing the features that are registered in vega's features library.

Copying a Features Library

You can copy a features library to and from LDM or tape using the COPY LIBRARY option. Select the COPY LIBRARY option from the LIBRARY MANAGEMENT menu on the Local Data Manager screen. The COPY LIBRARY property sheet appears as shown below.



Copying a Features Library to Another Library

1. Specify the name of the existing library in Library.
2. Choose To Lib and specify the name of a new library. Now the To Dir option becomes unhazed.
3. Choose To Dir and specify the directory where you want to save the Library.
4. Choose Apply.

Copying a Features Library from LDM

1. Choose From LDM and specify the LDM from which you want to copy a library. Now the To Dir and To Lib options become unhazed.
2. Specify the name of an existing library in the chosen LDM in Library.
3. Choose To Dir and specify the directory in which the new library has to be copied.
4. Choose To Lib and specify a new name for the library.
5. Choose Apply.

Copying a Features Library To Tarfile or Tape

1. Specify the name of library to be retrieved in Library.
2. Choose To Tape.
3. Choose Apply.

   The COPY LIBRARY command creates a file called `WRITE`-libname in the CADDS create directory.

   To copy the library to a tarfile, execute the `WRITE`-libname command in a shell tool:

```
% CADDS-create-dir/WRITE-libname
```

   `WRITE`-libname creates a tar file called libname`.tar`.

4. To copy the library to tape, edit the `WRITE`-libname file. This file contains two lines. The second line contains a default tarfile named libname`.tar`.

```
tar cvf libname.tar
```

   Change this tarfile name to a device name, such as /dev/rst0, for example,

```
tar cvf /dev/rst0
```

5. Execute the `WRITE`-libname command in a shell tool:

```
% CADDS-create-dir/WRITE-libname
```

Please note: The COPY LIBRARY command also creates a file called `sed_`libname in the directory where the features library libname resides. You must therefore have write permission in the directory that contains the features libraries.

Copying a Features Library from Tarfile or Tape

Specify the name of an existing library in Library.

1. Choose from Tape. Now the To Dir option becomes unhazed.
2. Choose To Dir and specify the destination directory.
3. Choose Apply.

   The COPY LIBRARY command creates a file called `READ-`libname in the CADDS create directory.

   Alternately, You can skip these three steps and enter the following command on the command line:

   ```
   Command: COPY LIBRARY NAME libname FROMTAPE
   DIRECTORY destination-directory
   ```

Please note: An existing library cannot be overwritten.

4. To copy a library from a tarfile, execute the `READ-`libname command in a shell tool:

   ```
   % CADDS-create-dir/READ-libname
   ```

5. To copy a library from tape, edit the `READ-`libname file. This file contains two lines. The first line contains a default tarfile named libname`.tar`.

   ```
   tar xvf libname.tar
   ```

   Change this tarfile name to a device name, such as /dev/rst0. For example,

   ```
   tar xvf /dev/rst0
   ```

6. Execute the `READ-`libname command in a shell tool:

   ```
   % CADDS-create-dir/READ-libname
   ```

Please note: The COPY LIBRARY command also creates a file called `edit_`libname in the directory where the features library libname resides. You must therefore have write permission in the directory that contains the features libraries.

Backing Up Features Libraries

Run the following script outside the application to back up a features library directory manually:

```
#!/bin/csh
source ~/.caddsrc
/usr/bin/dirname ` grep DATA_BASE_NAME $LDM_DIRECTORY | awk
'{print $5}'` > BKUP.LST
/usr/apl/cadds/scripts/sql SYS MGR << ! | grep / >> BKUP.LST
select pathname from part where part_name = 'FF-LIB%';
quit;
!
sed "s/ //g" BKUP.LST > FBM.LST
tar cvf FBM.BKUP -I FBM.LST
```

This script creates a file called `FBM.LST` that contains the directory path of the ldm directory (`/usr/ldm`) and the path names of all user-defined libraries.

Alternately, to find these path names within the application, run the LIST LIBRARY * command.

Updating the LDM with a New Revision

If you have the features database installed and you receive a new version of software, run the `ldmupdate` script to update the features system library. All user-defined features are preserved.

The `ldmupdate` script stops the features process to ensure that no user is currently working in the database. Then it updates representation parts or class definitions in the system library in the existing features database. Finally, it restarts the features process.

This procedure need be run only on previously installed LDM databases. For more information on the command, enter `ldmupdate -help`.

Run the command as follows:

```
# ldmupdate
```

Please note: If you have not run the SLIC (Software Loading Installation Command) program, you must also upgrade your LDM database. See *Installing CADDS 5* for information on the SLIC program.

Index

A

Accessing multiple libraries 2-16
Adding
  libraries 3-9
Additive features 2-21
  definition A-3
Applications
  Boolean 2-14
  description 2-10
  extend 2-13
  extended Boolean 2-14
  guidelines 2-15
  merge 2-13
  no application 2-15
Associating
  attributes 1-4
    menu option 3-18
Attaching properties 4-14
Attributes 1-4
  associating 1-4
    example in feature definition 3-24
    guidelines in creating representations 3-13
    menu option 3-18
  class 1-4
  conflicts of between superclasses 3-10
  constraints and 3-14
  disassociating 3-13, 3-18
  dynamic 1-4
    overwriting in feature definition 3-11
    restrictions on 3-4
  face 2-10
  in definition creation 1-4, 3-4
  inherited
    menu field for 3-6
  instance 1-4
  listing 2-18
    menu option 3-6
  menu field for changing 3-6
  names 3-4
  of properties 4-3
  of size 2-9, 3-5, A-7
  static 1-4
    menu field for 3-6
    restrictions on 3-4
  types of 3-4
  user dimensions and 3-14
  values of 3-4

B

Boolean application 2-14
  operands used with a 2-14
Bosses
  circular
    definition A-5
    property sheet 2-29
  definition A-5
Breaking associations
  between attributes and dimensions 3-13
Browse option 2-18
Browse property sheet 2-18
  in Teach Feature sessions 3-18
Browsing
  features, attributes, and libraries 1-5
  libraries

lists of 3-18
lists
in Teach Feature sessions 3-18
property sheet for 2-18

C

Change Feature option 2-47
Circular Boss option 2-29
Circular bosses
definition A-5
property sheet 2-29
Classes
basis of A-2
precedence of 3-9
Classifying features A-2
Command files
of representations 2-9
of user-defined features
example B-1
Conicalbottom Hole option 2-34
Conicalbottom Hole property sheet 2-34
Connectors A-3
Constraints
feature definitions and 3-14
example 3-25
Construction planes (Cplanes or CPLs)
instance insertion and 2-11
Copying
libraries 3-9
Counterbore Conicalbottom Hole option 2-37
Counterbore Conicalbottom Hole property sheet 2-37
Counterbore option 2-32
Counterbore Roundbottom Hole option 2-36
Counterbore Roundbottom Hole property sheet 2-36
Counterbored Hole option 2-30
Counterbores
attributes of size
illustration A-7
definition A-5
property sheet 2-32
Countersink option 2-33
Countersinks
attributes of size
illustration A-7
definition A-5
Countersinks property sheet 2-33
Countersunk Conicalbottom Hole option 2-37
Countersunk Conicalbottom Hole property sheet 2-37
Countersunk Hole option 2-31
Countersunk Roundbottom Hole option 2-36
Countersunk Roundbottom Hole property sheet 2-36
Creating
definitions 3-4
instances (features) 2-4
requirements 2-9
representations 3-13
user-defined features 3-2
Cutout option 2-35
Cutouts 2-23
definition A-6
property sheet 2-35

D

Default origins 2-11
Define Feature option 3-6
Define Feature property sheet 3-6
in Teach Features sessions 3-18
Defining
features 3-4
primitives 3-12
Definitions 1-4
attributes in
creating 1-4
requirements for 3-4
Boolean operations specified by 2-14
constraints in 3-14
example 3-25
creating 3-4
example 3-19
command file for B-1
property sheet 3-6
feature classes and 3-11
inheritance in 3-11
locking the representation of 3-2
modifying 3-2
multiple inheritance in 3-11
name length 3-4
names 3-4
overwriting of dynamic attributes 3-11
subclasses 1-4

superclasses 1-4
types of
subclasses 1-4
superclasses 1-4
user dimensions in 3-14
example 3-23
Delete Entity option 2-8
Deleting
definitions 3-18
instances 2-8
libraries 3-9
representations 3-18
Depressions
definition A-4
flats and grooves 2-24
slots 2-26
Disassociate Attributes option 3-13
Disassociating
attributes 3-18
Display Feature option 2-6
Displaying
features 2-6
Documentation, printing from Portable Document Format (PDF) file 1-xiv
Dynamic attributes
overwriting in feature definition 3-11
restrictions on 3-4

E

Entry faces
example 2-12
inserting instances and
in Extends 2-12
in Merges 2-13
Exit faces
example 2-12
inserting instances and
in Extends 2-12
in Merges 2-13
Extend application 2-13
restrictions with flats and grooves 2-25
Extended Boolean application 2-14
example 2-14

F

Face attributes 2-10
Feature field
in browsing 2-18
in Define Features property sheet 3-6
Feature libraries 1-5
accessing multiple libraries 2-16
adding a new library
property sheet for 3-9
browsing 1-5
browsing lists of 2-18
in Teach Feature sessions 3-18
copying to/from tape 3-9
deleting 3-9
listing 3-6
naming of features in 2-17
property sheet for selecting 2-18
in feature definition 3-6
selecting
in Teach Feature sessions 3-18
system library (SYSLIB) 2-16
Feature taxonomy A-2
Feature-based modeling
uses of 1-2
Features 1-3
creating
example 3-19
installing D-2
libraries
on client systems D-9
cross mounting D-10
local database manually D-4
shared database
setting up client systems D-6
testing shared database D-9
ldm_directory D-3
libraries D-2
local configuration D-4
on client systems D-9
cross mounting D-10
local configuration D-4
name length 3-4
shared database D-6
advantages D-6
starting
automatically D-10

restarting with ldmserv D-10
troubleshooting D-11
viewing 2-19
Features menu
Browse option 2-18
Change Feature option 2-47
Circular Boss option 2-29
Conicalbottom Hole option 2-34
Counterbore Conicalbottom Hole option 2-37
Counterbore option 2-32
Counterbore Roundbottom Hole option 2-36
Counterbored Hole option 2-30
Countersink option 2-33
Countersunk Conicalbottom Hole option 2-37
Countersunk Hole option 2-31
Countersunk Roundbottom Hole option 2-36
creating a definition with 3-5
Cutout option 2-35
Define Feature option 3-6
Delete Entity option 2-8
Flat option 2-34
Hole option 2-29
Insert Feature option 2-4
Rectangular Groove option 2-33
Rectangular Pockets option 2-31
Rectangular Slot option 2-32
Rectangular Tab option 2-30
Roundbottom Hole option 2-35
Select Library option 2-18
creating a definition using 3-6
system feature options on 2-28
Undo Operation option 2-8
Verify Feature Instance option 2-43
Features troubleshooting
application session C-2
defining features C-4
modeling C-3
FFIM A-1
Flat option 2-34
Flats 2-24
attributes of size
illustration A-7
definition A-6
example 2-25
property sheet 2-34
restrictions with Merge or Extend 2-25
Form features A-3
diagram of A-7
types of
additive features A-3
subtractive features A-4

G

Grooves 2-24
attributes of size
illustration A-7
definition A-6
example 2-26
property sheet 2-33
restrictions with Merge or Extend 2-25

H

Hazed areas
on property sheets 3-7
Hole option 2-29
Holes
attributes of size
illustration A-7
list of
passage holes 2-24
subtractive holes 2-23
property sheets
conicalbottom holes 2-34
counterbored 2-37
countersunk 2-37
counterbored holes 2-30
conicalbottom 2-37
roundbottom 2-36
countersunk holes 2-31
conicalbottom 2-37
roundbottom 2-36
holes 2-29
roundbottom holes 2-35
counterbored 2-36
countersunk 2-36

I

Inheritance
with properties 4-4
Insert Feature option 2-4
Insert Feature property sheet 2-4

Installing
Features D-2
installing local database manually D-4
installing shared database
setting up client systems D-6
testing D-9
libraries
on client systems D-9
cross mounting D-10
starting
automatically D-10
restarting with ldmserv D-10
troubleshooting D-11
Instances 1-3
applications 2-10
Boolean 2-14
extend 2-13
extended Boolean 2-14
guidelines 2-15
merge 2-13
no application 2-15
creating 2-4
definition 1-3
deleting 2-8
displaying 2-6
inserting 2-4
attributes required for 2-9
description 2-2
example 2-38
property sheet 2-4
undoing an insertion 2-8
locations of 2-11
modifying
by attributes 2-47
by parameters 2-51
restrictions on 2-50
orientation of 2-11
origins of 2-11
parametric aspects of
in deleting instances 2-8
in modifying instances 2-51
removing 2-8
viewing 2-4
Interactive View Controls toolbar 2-6
Intervening faces
used in Extended Boolean application 2-14
example 2-14

L

Library entries
precedence of 2-17
Library options 2-18
Listing
attributes 3-6
Lists of libraries
menu for browsing 2-18
Local Data Manager (LDM) 3-4
Local operations
inserting instances and 2-13
Location
of instances 2-11
Locking representations 3-2
in Teach Feature sessions 3-15

M

Material condition attribute item 3-5
Merge application 2-13
restrictions with flats and grooves 2-25
Modifying
attributes 3-6
instances (features)
by attributes 2-47
by parameters 2-51
user-defined features
definitions of 3-2
representations of 3-3
Modifying properties 4-3
Multiple inheritance 3-9
definitions and 3-11
description 1-4
diagrams for determining A-7
listing inherited attributes 3-6
precedence of classes in 3-9
subclasses, superclasses and 1-4
Multiple libraries
accessing 2-16

N

Naming
attributes 3-4

features 3-4
overriding of 2-17
NoApply application 2-15

O

Operands in Boolean applications 2-14
Orientation
of instances 2-11
Origins
default 2-11
Overriding of feature names
between libraries 2-17
Overwriting
dynamic attributes 3-11

P

Parametric histories
instance insertion captured by 2-8
representations and 1-4
Parametrically Controlled Entities 3-14
Parametrics
changes requiring model regeneration 2-51
instance modification and 2-51
Passages
cutouts 2-23
thru holes 2-23
PDES A-1
Pockets
attributes of size
illustration A-7
definition A-5
property sheet 2-31
Precedence
of library entries 2-17
Primitives A-4
defining 3-12
Printing documentation from Portable Document Format (PDF) file 1-xiv
Properties
attaching 4-14
attribute values 4-15
attributes 4-3
defined 4-2
libraries 4-4
tagnames 4-15
use of inheritance 4-4
using 4-2
Property instance condition 4-20
Property sheets
hazed areas 3-7
Protrusions A-3

R

Rectangular Groove option 2-33
Rectangular Groove property sheet 2-33
Rectangular Pockets option 2-31
Rectangular Pockets property sheet 2-31
Rectangular Slot option 2-32
Rectangular Slot property sheet 2-32
Rectangular Tab option 2-30
Rectangular Tab property sheet 2-30
Regenerating models
after changing parameters 2-51
during Teach Feature sessions 3-26
Removing
instances 2-8
Representation
example 3-21
Representations 1-4
attribute association in 3-13
command files of 2-9
copying 3-17
creating 3-13
example 3-19
command file for B-1
guidelines on attribute association 3-13
locking 3-2
in Teach Feature sessions 3-15
modifying 3-3
orienting
in Teach sessions 3-18
Teach Feature sessions and 1-4
Restrictions
on modifying instances 2-50
Retrieving
instance information 2-43
Roundbottom Hole option 2-35
Roundbottom Hole property sheet 2-35

S

Select Library option 2-18
  creating a definition using 3-6
Select Library property sheet 2-18
  in defining features 3-6
  in Teach Features sessions 3-18
Selected Library field
  in Define Features property sheet 3-6
Selecting
  feature libraries 2-18
    property sheet 3-6
  system features 2-4
Size attributes 2-9
  illustrations of A-7
  material condition and 3-5
  units of 3-5
Slots 2-26
  attributes of size
    illustration A-7
  definition A-6
  example of a full slot 2-27
  example of a partial slot 2-28
  property sheet 2-32
Static attributes
  menu field for 3-6
  restrictions on 3-4
Subclasses 1-4
  diagrams for determining A-7
  precedence of classes 3-9
  system features and 3-11
Subtractive features 2-22
  definition A-4
Superclasses 1-4
  attribute inheritance and 3-6
  conflicts among 3-9
  diagrams for determining A-7
  Feature superclass 3-11
  menu field for 3-6
  precedence of classes 3-9
  system features and 3-11
SYSLIB (System library) 2-16
  user-defined libraries and 2-16
System Feature options 2-28
System features
  additive 2-21
    definition A-3
    list of 2-22
      Boss_Circular 2-22
      Tab_Rectangular 2-22
  categories of 2-21
    additive 2-21
    subtractive 2-22
  classification of A-2
  definition A-2
  depressions
    flats and grooves 2-24
    slots 2-26
  descriptions of A-5
  diagrams of classes A-7
  flats 2-24
  grooves 2-24
  inserting 2-9
    example 2-38
  library (SYSLIB) of 2-16
  passages 2-23
    list of 2-24
  selecting 2-4
  setting tolerance value 2-19
  size attributes of
    illustrations of A-7
  slots 2-26
  subtractive 2-22
    definition A-4
    list of 2-23
      Counterbore 2-23
      Countersink 2-23
      Hole_Blind 2-23
      Hole_Cbored 2-23
      Hole_CboredConical 2-23
      Hole_CboredRound 2-23
      Hole_ConicalBottom 2-23
      Hole_Csunk 2-23
      Hole_Csunkconical 2-23
      Hole_Csunkround 2-23
      Hole_RoundBottom 2-23
      Pocket_Rectangular 2-23
  superclass Feature and 3-11
  thru holes 2-23
System library (SYSLIB) 2-16
  user-defined libraries and 2-16

T

Tabs

definition A-5
property sheet 2-30
Teach Feature sessions 1-4
activating 3-17
browsing lists in 3-18
creating representations in 3-13
example 3-19
command file for B-1
exiting 3-15
parametrically controlled entities 3-14
regenerating models during 3-26
representations in 1-4
tasks required in 3-13
Thru holes 2-23
Types
of attributes 3-4

U

Undo Operation option 2-8
Undo Operation property sheet 2-8
Undoing
instance insertions 2-8
parametric operations 2-8
Unlocking
representations 3-15
User dimensions
feature definitions and 3-14
example 3-23
User-defined features
classification of A-3
creating 3-2
example 3-19
command file for B-1
User-defined libraries
system library and 2-16

V

Values
of attributes 3-4
Verify Feature Instance option 2-43
Verifying
instance information 2-43
View Feature mode 2-4
Viewing
features 2-4
Virtual libraries 2-16
Voids
definition A-4
Volume features A-3

W

Workpieces 1-3
inserting instances and 2-10